FEBRUARY 1985 VOL. 3, NO. 2 $3.95

FOR IBM PERSONAL COMPUTER USERS

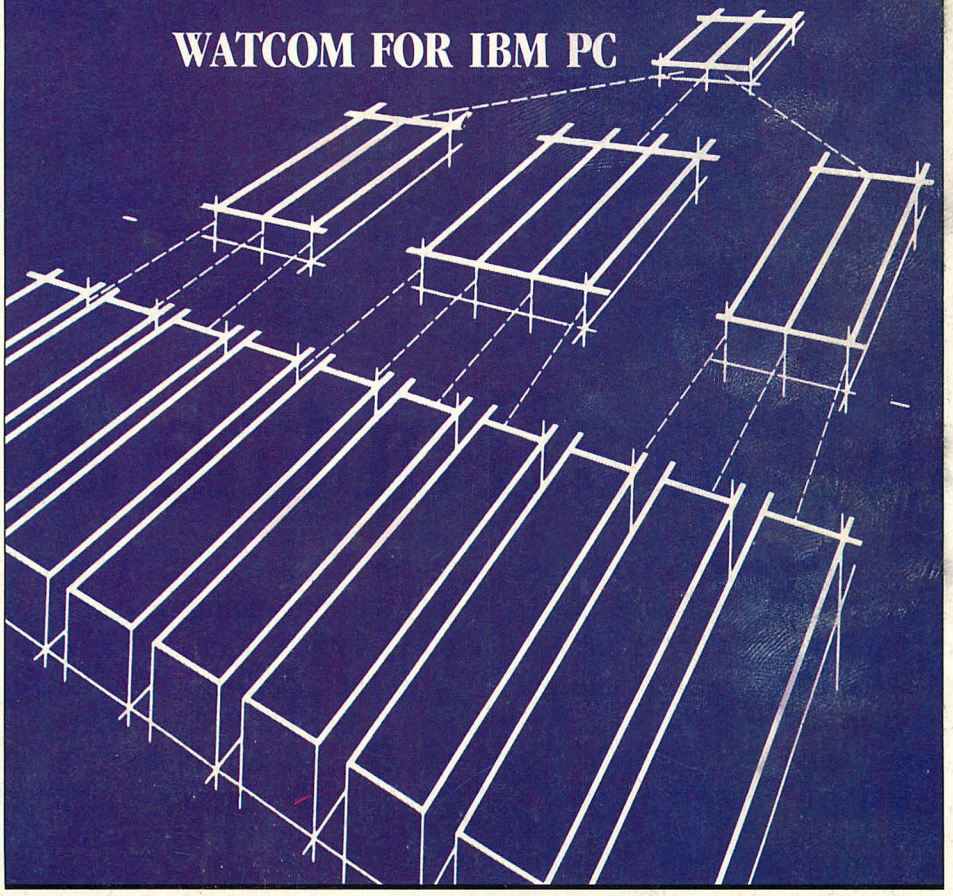
TECH JOURNAL

UNDERSTANDING B-TREES
Efficient Solutions for Data Retrieval

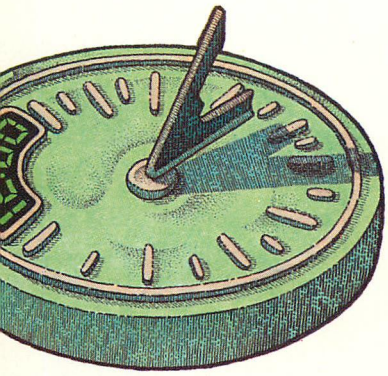
EXTENDING TURBO PASCAL

MEMORY OPTIONS FOR PC*jr*

WATCOM FOR IBM PC

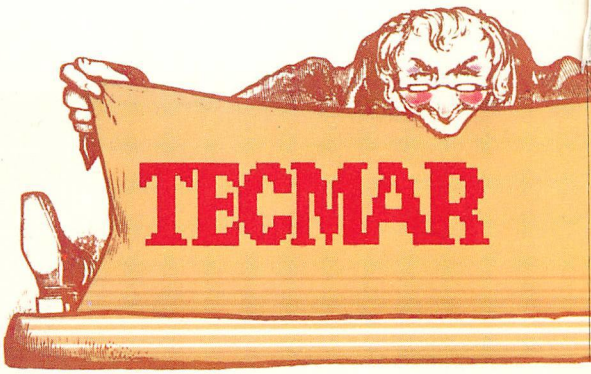


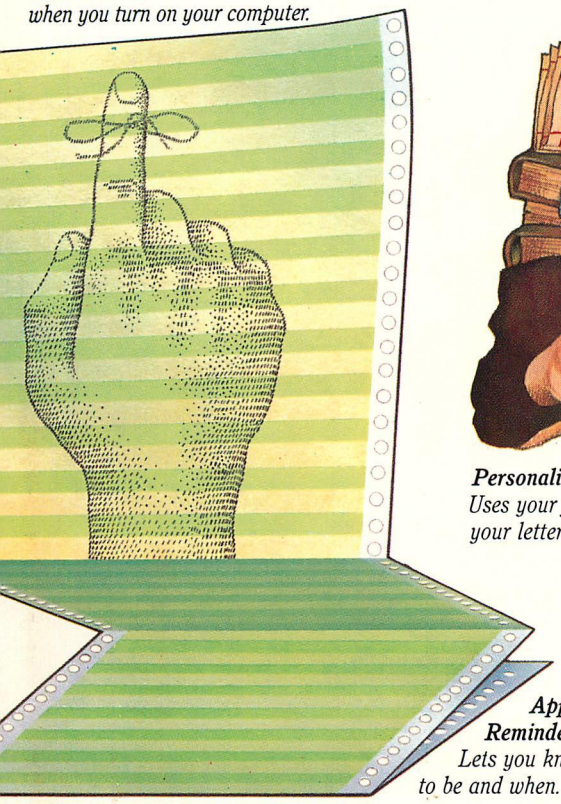
02
0 14024 14357



Clock/Calendar
Automatically inserts the date and time when you turn on your computer.

Encode/Decode
Ensures the security of your files.

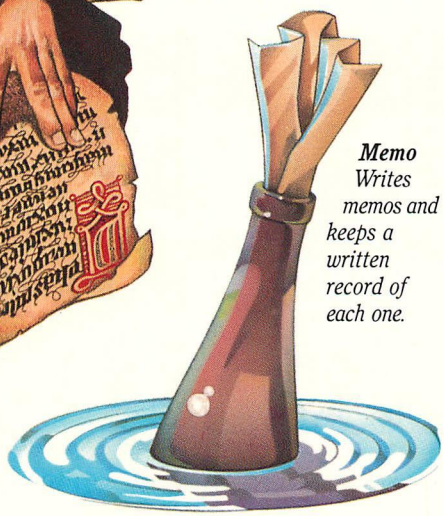




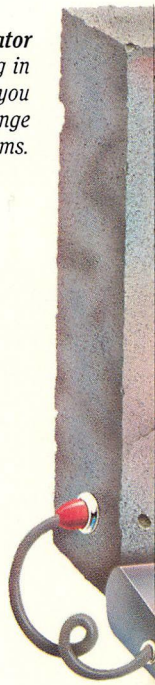
Appointment Reminder
Lets you know where to be and when.

Personalized form letters
Uses your file to personalize your letters and make labels.

Calculator
Does your figuring in background mode so you won't have to change programs.



Memo
Writes memos and keeps a written record of each one.



Tecmar thinks your PC

As your business grows, so should your computer. Only Tecmar offers you this unique package of hardware and free software. With it your PC can grow to its full potential.

Tecmar Multifunction Boards...

"extremely powerful."—PC World

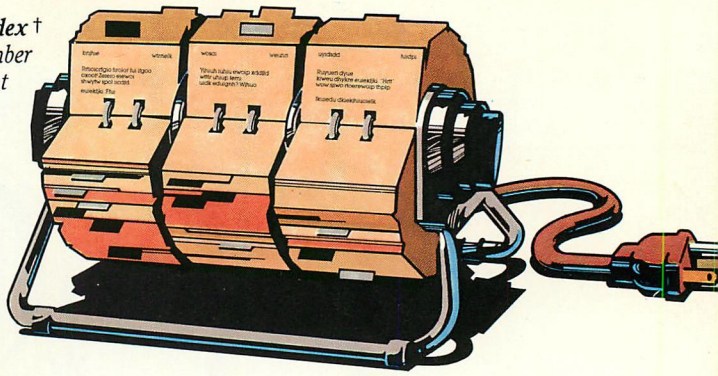
A Tecmar board will expand your computer's memory and increase its speed and efficiency. It will give you additional ports to accommodate a wide range of peripherals from modems and plotters to dot matrix and letter quality printers.

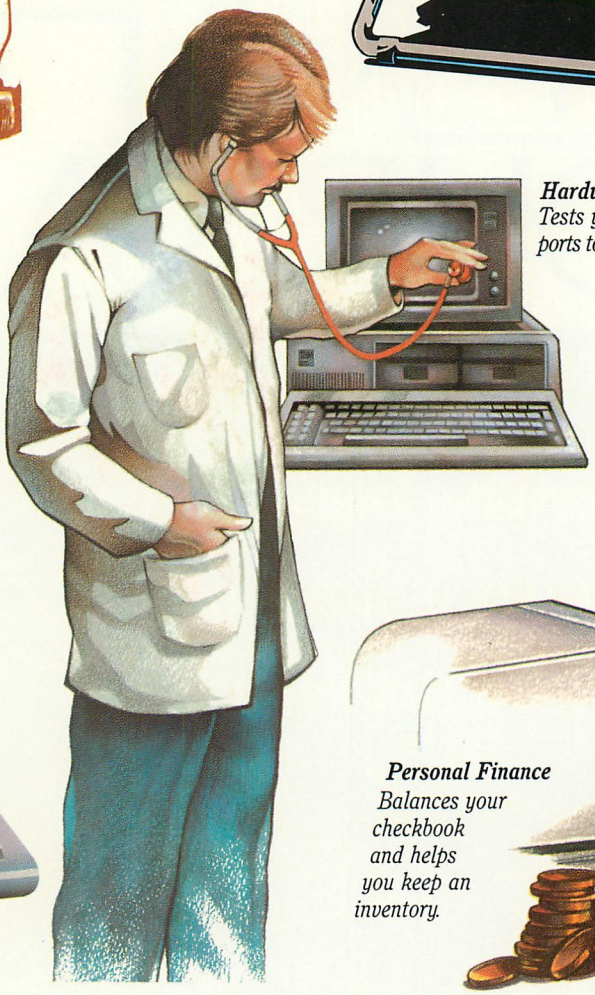
With a Tecmar board you can run powerful programs like Lotus 1-2-3™* and WordStar®†† on a PCjr. The RamSpooler makes printing a background task and frees your PC for other jobs. A built-in clock/calendar automatically inserts the date and time at power on and is independently powered by an easily replaced battery.

Every Tecmar multifunction board is run through a series of rigorous tests to ensure quality. Our incredibly low failure rate (0.4%) is unparalleled. All boards are additionally backed by a full one-year warranty.

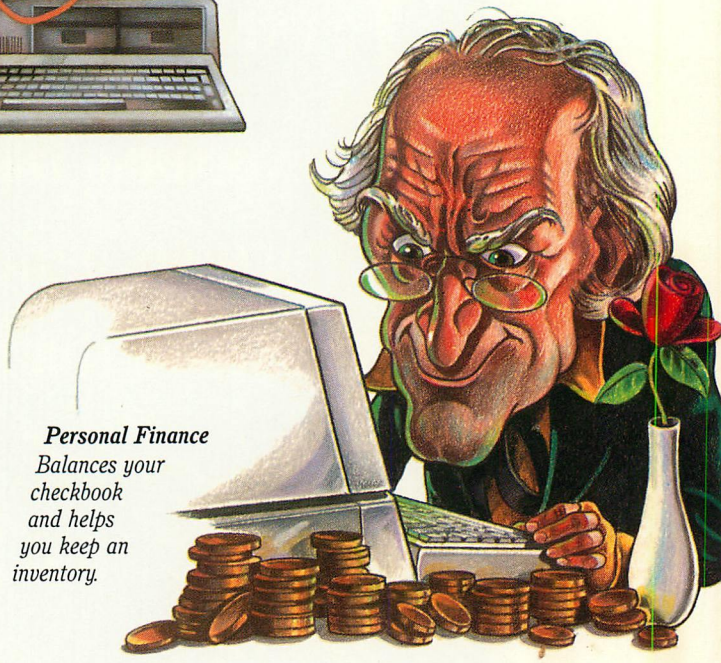
Banner
Allows you to print in a variety of type sizes. Even large banners!

***Rolodex*†**
Finds the name and number you need instantly without changing programs.





Hardware Diagnostics
Tests your PC's memory and parallel ports to stop problems before they happen.

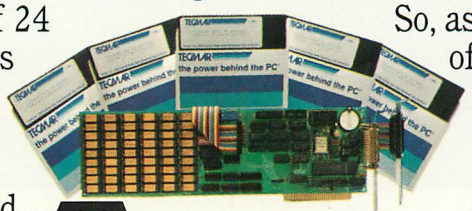


Personal Finance
Balances your checkbook and helps you keep an inventory.

should do all of the above.

Free Software *"...a chest of Jewels."—PC Week*
Great hardware deserves great software. So, if you buy a Tecmar board we'll give you our Treasure Chest™ of Software at no extra charge. The Treasure Chest consists of 24 programs that include business applications, a calculator, a security system, hardware diagnostics, even games! Most of these can be run in background mode with programs like Lotus 1-2-3 and WordStar. Using these features is as easy as a couple of keystrokes, and without changing disks. No other company offers you such an extensive array of software with their multifunction boards.

So, ask your dealer for a demonstration of any of Tecmar's multifunction boards. And check out the free software while you're there. Or call 216-349-0600 for the dealer nearest you.



Tecmar
THE POWER BEHIND THE PC

*Lotus 1-2-3 is a registered trademark of the Lotus Development Corporation. †Rolodex is a registered trademark of the Rolodex Corporation. ††WordStar is a registered trademark of the MicroPro International Corporation.

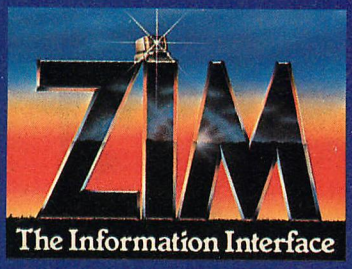
CIRCLE NO. 187 ON READER SERVICE CARD

The power behind your PC is right behind this page.

The multi-user data base management system that puts mainframe power in your micro.

ZIM is a fully integrated fourth generation application development language that is loaded with these features and more.

- POST RELATIONAL
  - — Entity Relationship Model
  - — Powerful extension of Relational Model
- REPORT WRITER
  - — Unlimited break levels, summary/ detail reports, output to disk, printer, terminals
- FORMS PAINTER AND MANAGER
  - — Menus, data entry, data display
  - — Box fields
  - — Old field value recall
- DATA DICTIONARY
- COMPILER • PROGRAMMING LANGUAGE
- APPLICATION COMPLEXITY SUBJECT ONLY TO HARDWARE LIMITATIONS
- UNLIMITED FILE RELATIONS
  - — One to one
  - — One to many
  - — Many to many
  - — Unrelated



- RETRIEVAL STRATEGY OPTIMIZER
  - — Automatic use of 8087 chip (if available)
- APPLICATIONS PORTABILITY
- MULTI-USER
  - — Full transaction processing control
- C LANGUAGE INTERFACE
- QUALITY PRODUCT SUPPORT

Zim is a mainframe data base management system that runs on micro-computers. If you want mainframe power, speed, flexibility and freedom from arbitrary limitations all at a micro price, talk to us about an evaluation system.

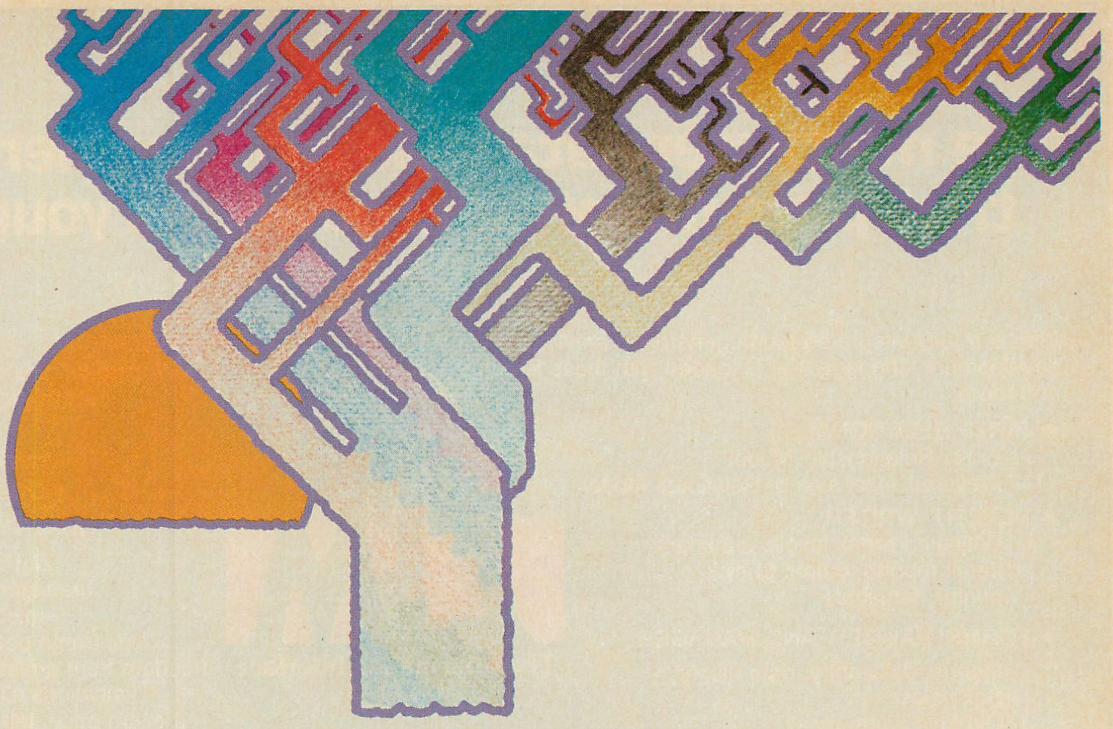
Dealer inquiries are welcome.

ZANTHE INFORMATION INC.

1785 Woodward Drive
Ottawa, Ontario K2C 0R1
(613) 727-1397

OUR FEATURE STORY

CIRCLE NO. 148 ON READER SERVICE CARD



Btrieve and Xtrieve. B-tree file access for all your programming languages. With the query tool your users demand.

Introducing a powerful data base combination for PC application developers. Btrieve™, the most sophisticated file access method for your IBM™ PC. In single user and network versions. And Xtrieve™, a new menu-driven query tool that gives you—and your users—fast access to information.

Btrieve: for professional programmers. Btrieve provides fast, flexible file management for all your application development. All your programming languages—BASIC, Pascal, Cobol, C. With multikey access to records. Automatic file recovery. Unlimited records per file. Duplicate, modifiable, and segmented keys. Variable cache buffer.

Better applications faster. Based on the b-tree file indexing system, Btrieve provides optimal file management and eliminates writing file management routines. So you can develop *better* applications *faster.*

Xtrieve: easy window interface. The ideal complement to Btrieve, Xtrieve is the non-programmer's interface. Xtrieve's full relational capabilities let users define a virtual table of data from multiple files. Then Xtrieve speeds them through query building with a series of easy-to-follow windows.

No command language. Xtrieve is completely menu-driven, so there's no need to memorize command language. Or special syntax. Everything you need is on the screen. You and your users can perform unlimited queries—and get answers fast.

Xtrieve features a full range of restriction criteria. Online help messages. Automatic maintenance of information order by multiple indexes. And interfaces to access information from Lotus 1-2-3™ and dBase II® files.

Network capabilities. Network versions of Btrieve and Xtrieve allow data sharing in the most popular PC networks — PCnet™, NetWare™, EtherSeries™, MultiLink™, and OmniNet™.

For more information or to order, call or write:

SoftCraft Inc.

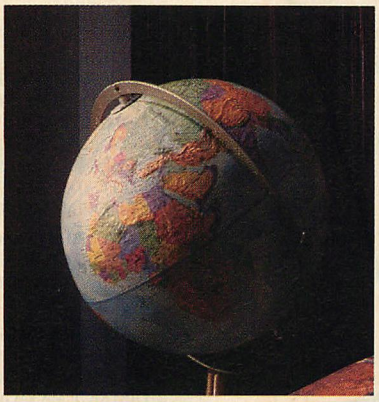
P. O. Box 9802 #590 Austin, Texas 78766 (512) 346-8380

Suggested retail prices: Btrieve, $245; Xtrieve, $195. Btrieve/N (network), $595; Xtrieve/N, $395. Dealer inquiries welcome. Btrieve requires PC-DOS or MS™-DOS 1.X or 2.X; Xtrieve, PC-DOS or MS-DOS 2.X.

TECH JOURNAL

FEBRUARY 1985 VOLUME 3, NUMBER 2

102



126

150

PC TECH JOURNAL

VOL. 3, NO. 2

PUBLISHER: *Newton Barrett*
EDITOR: *Will Fastie*

EDITORIAL
MANAGING EDITOR: *Marjory Spraycar*
EXECUTIVE EDITOR: *Julie Anderson*
TECHNICAL EDITOR: *Susan Glinert*
SENIOR COPY EDITOR: *Susan Holly*
COPY EDITOR: *Gail Shaffer*
CONSULTING EDITOR: *Thomas V. Hoffmann*
CONTRIBUTING EDITORS: *Ray Duncan, Richard Foard, Arthur A. Gleckler, Augie Hansen, William H. Murray, Max Stul Oppenheimer*
EDITORIAL SECRETARY: *Diana L. Carey*
EDITORIAL ASSISTANT: *Carole Autenzio*

ART & PRODUCTION
ART DIRECTOR: *Nancy Lepow*
ASSISTANT ART DIRECTOR: *Jane Frey*
ART ASSISTANT: *Sandra Ray*

ADVERTISING SALES
MARKETING COORDINATOR: *Julie Henderson*
ADVERTISING COORDINATOR: *Michele Fischetti*
DISTRICT MANAGERS:
Rita Burke, Ian Smith—East Coast
Lisa Kampfmann—Midwest
Ted Bahr, Bill Bush, Harriet Rogers—West Coast
ACCOUNT REPRESENTATIVES:
Jan Schultz—East Coast
Rosemarie Caruso—Midwest
Pam Sigal, Arlene Steadman—West Coast

CIRCULATION
SUBSCRIPTION DIRECTOR: *Chet Klimuszko*
DIRECT MAIL MANAGER: *Eric A. Bernhard*
RENEWAL AND BILLING MANAGER: *Shane Boel*
MEDIA MANAGER: *Ann Pichette*
SPECIAL SUBSCRIPTION SALES: *Daniel Rosensweig*

COMPUTER PUBLICATIONS DIVISION
PRESIDENT: *Kenneth H. Koppel*
SENIOR VICE PRESIDENT: *Eileen G. Markowitz*
VICE PRESIDENT, Editorial: *Jonathan D. Lazarus*
VICE PRESIDENT, Production: *Baird Davis*
VICE PRESIDENT, Licensing & Special Projects: *Jerry Schneider*
VICE PRESIDENT, Creative Services: *Herbert Stern*
VICE PRESIDENT, Circulation: *Alicia Marie Ivans*
MARKETING MANAGER: *Ronnie Sonnenberg*
BUSINESS MANAGER: *Gary A. Gustafson*
EDITORIAL DIRECTOR: *Ernest F. Baxter*

ZIFF-DAVIS PUBLISHING
PRESIDENT, *Richard P. Friese;* SENIOR VICE PRESIDENT, *Philip T. Heffernan;* SENIOR VICE PRESIDENT, *Philip Sine;* VICE PRESIDENT, *William L. Phillips;* TREASURER, *Selwyn Taubman;* SECRETARY, *Bertram A. Abrams*

EDITORIAL OFFICE
PC TECH JOURNAL, The World Trade Center, Suite 211, Baltimore, MD 21202. 301/576-0770. The Source ID STY682.

ADVERTISING OFFICES
(East Coast) One Park Avenue, New York, NY 10016. 212/503-5100.
(Midwest) 180 N. Michigan Avenue, Chicago, IL 60601. 312/346-2600.
(West Coast) 11 Davis Drive, Belmont, CA 94002. 415/594-2290.

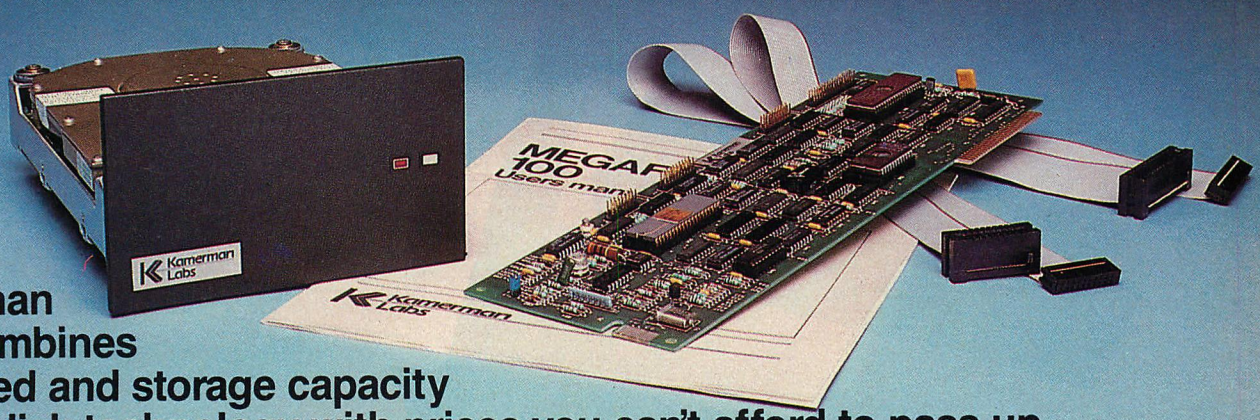
SUBSCRIPTION INQUIRIES
PC TECH JOURNAL, P.O. Box 2968, Boulder, CO 80321. Subscription service: 800/631-8112. Back issues: send $7.00/copy to *PC TECH JOURNAL,* Box CN, 1914, Morristown, NJ 07960.

PC TECH JOURNAL (ISSN 0738-0194) is published monthly, $29.97 for one year, $52.97 for two years, $69.97 for three years. Additional postage $12 for Canada & Foreign by Ziff-Davis Publishing Company, One Park Avenue, New York, NY 10016. Application for Second-Class Postage Rates pending at New York, NY 10001. POSTMASTER: Send address changes or subscription inquiries to P.O. Box 2968, Boulder, CO 80321.

FOR THE IBM® PC, IBM® XT, COMPAQ,™
CORONA,™ CORONA™ PORTABLE, LEADING EDGE,® EAGLE PC,® TAVA.™

10 Mbyte Hard Disk System

~~$895~~ $695



Kamerman Labs combines the speed and storage capacity of hard disk technology with prices you can't afford to pass up.

A dramatic leap in performance for your personal computer.

Even the smallest Kamerman Labs hard disk unit gives you 30 times the storage capacity of a typical floppy disk. Sophisticated business and professional data storage requirements can now be handled with ease.

Better yet, you can access data up to 10 times faster than with floppies. Spend more time working and less time waiting.

And it's easy to use. Kamerman Labs lets you boot directly off the hard disk, just like the IBM XT.

A complete backup solution.

Backup of hard disk data is a critical requirement in many microcomputer applications, so Kamerman Labs offers several reliable and cost-effective backup devices that are fully compatible with our hard disk units.

Lower prices than ever.

Because you deal directly with Kamerman Labs, we offer you remarkably low prices. Finally, you can realize the full benefits of hard disk technology without the excessive costs.

Tested and formatted.

All Kamerman Labs systems have been thoroughly tested and formatted before delivery.

Full warranty and technical support.

Kamerman Labs backs all its products with a full one-year warranty on both parts and labor. Plus you get prompt technical support through a toll-free hotline number.

To order, call toll-free 800-522-2237.

In Oregon, Hawaii and Alaska call **503-626-6877.**

All Kamerman Labs Units come in either internal or external configurations.

IBM is a registered trademark of International Business Machines Corp.
Compaq is a trademark of Compaq Computer Corp.
Corona is a trademark of Corona Data Systems.
Leading Edge is a registered trademark of Leading Edge Products, Inc.
Eagle is a registered trademark of Eagle Computers.
TAVA is a trademark of TAVA Corporation.

Kamerman Labs

8054 S.W. Nimbus, Bldg. 6, Beaverton, Or 97005
Phone 503-626-6877

CIRCLE NO. 237 ON READER SERVICE CARD

Picture a computer under $1000 that can run Lotus 1-2-3.

To run a powerful program, you need a powerful computer. But "powerful" doesn't always have to mean expensive.

Case in point: PC*jr*, from IBM.

With its 128KB memory, PC*jr* can run the world's best-selling business program—Lotus 1-2-3—in its new cartridge form. Giving you the power to integrate spreadsheets and data bases, and visualize numbers in charts and graphs.

PC*jr*'s cartridge format offers some real advantages, too.

A cartridge not only loads much faster than a program on diskette—it uses almost no user memory. So you get more "room to work."

It can also free the diskette drive to be used for information storage alone.

And perhaps best of all, a cartridge program can't be erased. Which means your investment is safe.

Of course, PC*jr* runs diskette programs as well. Over a thousand of the best programs written for the IBM PC. For business, home management, communications, education and entertainment.

And for all of its power, it costs less than $1,000,* without monitor.

"But I already have Lotus™ 1-2-3™ on diskette."

If that's the case, you may not want to buy the cartridge version. All you need is a PC*jr* Installation Kit (available free where you bought 1-2-3) and the new 128KB PC*jr* Memory Expansion Attachment.

This doubles PC*jr*'s memory. And, by no coincidence, it also doubles the number of programs you'll be able to run.

So you can use Lotus 1-2-3 on diskette, and over a thousand *additional* programs that utilize expanded memory.

Picture yourself with PC*jr*. You can try one out at an authorized IBM PC*jr* dealer or IBM Product Center.

For the name of the store nearest you, call 1-800-IBM-PCJR. In Alaska and Hawaii, call 1-800-447-0890. IBM®

IBM PC*jr*

Growing by leaps and bounds.

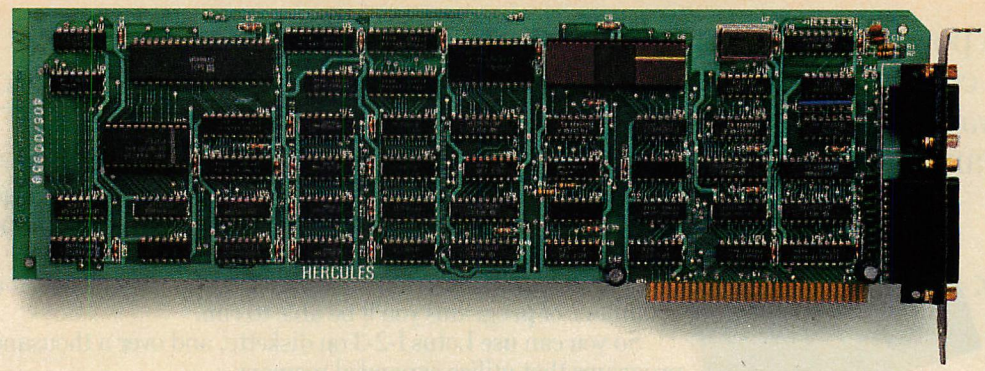
Before or after you buy a PC*jr*, it's easy to get answers to your questions. Just call 1-800-222-PCJR.

*IBM Product Center price, monitor not included.

Lotus and 1-2-3 are trademarks of Lotus Development Corp.

Little Tramp character licensed by Bubbles Inc., s.a.

CIRCLE NO. 152 ON READER SERVICE CARD



Introducing the Hercules™ Graphics Card for the technical user.

OK. We confess. The Hercules Graphics Card in the picture above isn't a special version for the technical user.

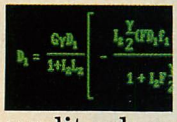
In fact, it's exactly the same as the standard Hercules Graphics Card running programs like 1-2-3™ and Symphony™ in more than 100,000 IBM® PCs.

We just wanted to make the point that the Hercules Graphics Card is not only big with business users—it's also the most popular high resolution graphics card for the technical user.

Why? We run more software than anyone else.

The Hercules Graphics Card is supported by more technical software than any other hi-res graphics card.

There are word processors that can produce publication quality documents with mathematical formulas.



There are programs that enable your PC to emulate a graphics terminal and run mainframe graphics software.

There are toolkits of graphics utilities that can be linked to popular programming languages.

There are CAD programs that can provide features normally associated with $50,000 systems.

And we supply free software with each card to do hi-res graphics with the PC's BASIC. No one else does.

Hardware that set the high performance standard.

When we introduced the Hercules Graphics Card in August, 1982, it set the standard for high resolution graphics on the PC.

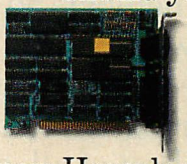
But we didn't stop there. In the past two years, we've continually refined the original design.

Today's Graphics Card gives you two graphics pages, each with a resolution of 720h x 348v, and a parallel printer port—standard.

A 2K static RAM buffer elegantly eliminates scrolling flicker. And our exclusive safety switch helps prevent damage to your monitor.

Convinced? Good. Now, how about a little color?

Should you want IBM compatible color graphics for your system, then the new Hercules Color Card is the smart way to go.



It gives you a parallel printer port and a size small enough to fit in one of the XT's or *Portable's* short slots.

And both Hercules cards are compatible with the new AT™ and backed by our two year warranty.

Call 800 255-5550 Ext. 403 for the name of the Hercules dealer nearest you and we'll rush you a free info kit. See why the company that made the first graphics card for the IBM PC still makes the best.

Hercules.
We're strong on graphics.

Address: Hercules, 2550 Ninth St., Berkeley, CA 94710 Ph: 415 540-6000 Telex: 754063 **Trademarks/Owners:** Hercules/Hercules Computer Technology, 1-2-3, Symphony/Lotus Development; IBM, AT/International Business Machines

CIRCLE NO. 133 ON READER SERVICE CARD

The Compatibility Trap

Will IBM be compatible with IBM?

IBM has done a magnificent job with compatibility. The PC family's five members (PC, XT, Portable, PC*jr*, and AT) are so similar that most software will run on any of the machines. If a program will not work at first the required changes are usually minimal. IBM's attention to compatibility has been so good that the PC*jr*, for example, can execute programs, which write directly into video memory, at memory addresses not physically present in the machine. The AT does an excellent impression of an XT, even to the point of accepting 8-bit adapter boards.

The IBM mainframe community has come to expect this. An MIS department makes a massive investment in software, both purchased or written in-house, and hardware, whether from IBM or a compatible vendor. Once that money has been spent and an operational posture attained, MIS will not be willing to dispose of its solutions simply because the foundation of its strategy, the computer itself, has changed. What MIS would prefer to do is ask for a new machine just like the old one but [fill in your favorite incremental performance factor here] times faster. In such a request, compatibility with the previous generation is implicit.

Such long-term compatibility has won IBM many friends and even more customers. In more modern times, many of us have become endeared as we have watched the compatibility evolution of the PC family. Now, however, I believe IBM is about to reach critical mass and introduce new machines into the family that are not fully compatible with their siblings. Why would IBM do such a thing? How can it possibly be in its best interest to do so? The answer is chillingly simple: it is cheaper.

IBM is engaged in a very fierce marketing battle these days. Competition in pricing is tough, and IBM must build products with reasonable (for IBM) margins that are still competitive in the open market. The marketing department therefore wants the prices to drop, yet, maintaining compatibility is expensive. Consider the case of the new IBM enhanced graphics adapter. This board supports every text and graphics mode heretofore defined by IBM for the PC, as well as the new, extended modes. At the very least, that implies a complex, new BIOS routine for the video service interrupt, a routine that must be delivered to the host PC via a ROM chip on the new board. The same thing applies to the hardware. Anything the old boards could do, this new board must be able to do too, and more.

When we, the end users, buy the new board, we will be pleased that IBM has not ignored the history of the PC in providing all the necessary features. I, for instance, will be able to replace both my monochrome and color boards with a single, enhanced board and still have all the capability (and a little more) than I did before. I can add the new display and gain even more. Still, though, I will remain compatible.

A COMPATIBLE FUTURE?

Now, consider the situation two years down the road: suppose some large number of enhanced color systems have been installed, and there are, still, a great many monochrome and old color systems. IBM determines that a new standard could be put forward (how about 768-by-512 pixels, or maybe 1,024 pixels square, or perhaps 256 colors) that is cost-effective, technologically feasible, possibly faster, and desirable from a marketing point of view. Will it come with all the old modes?

I think not. The point to remember is that a new board with all the old modes carries a lot of excess baggage. As time goes on, most users will take advantage of only the newest modes. Their software will be upgraded to support those options and programs will have appeared that take full advantage of the new functions. In effect, they will, have migrated to a new technology. Yes, many probably will use a machine elsewhere (in their home, on a trip) that does not have the new modes, but they likely will use it in a text-only mode, something all the IBM machines will support easily. The old graphics modes become painful to support, expensive to build, and a general nuisance. Sooner or later, IBM will drop them. When they do, the machine for which they are dropped will not be compatible with previous members of the family.

Graphics is not the only item on the compatibility chart. Another example is diskette drives and their media. Today, IBM supports three types of 5¼-inch diskette drives: single-sided, double-density; double-sided, double-density; and the new high density. The original single-sided drives are really artifacts. With the open-market price of double-sided drives at less than $150, no one would seriously consider the smaller, more limited unit. Maintaining support for the smaller unit is not expensive in a software sense, but it is a complicating factor. For example, the number of options in programs such as FORMAT continues to rise, making it more difficult for a user to comprehend. In the old days (that's three years ago), one simply said FORMAT, not FORMAT /8/1. This is a simple example, but every time IBM announces a new disk storage option, its programmers must make sure that all such programs are adapted and extended as required. More complexity, more programming, more documentation, more time, more confusion. And more cost.

The rise in complexity is marked by the arrival of the AT and its SETUP program. System configuration DIP switches on the system board are okay to a point, but, to be consistent, there would have had to have been four banks or so on the AT. IBM opted for the soft approach, and wrote a program that holds the user's hand, leading him through the configuration procedure. Right now, that program is simpler to

use and easier to understand than the switches on the original PC. But what about next year? At the rate IBM is announcing new products, the SETUP program could take four times as long and some answers might be dependent on others, complicating matters further still.

But confusion of the end user is not the real problem. Of greater concern is the difficulty third-party software developers will have in deciding how to build their systems. Which color mode (or modes) should be supported? Which system configurations? How can the program tell? What steps should a program take if it cannot come to a conclusion that is anything except ambiguous about what the machine looks like? If multiple options are to be supported, how is the program to tell which the user would prefer?

My conclusion—that IBM will soon announce machines and operating environments that do not support all of the options of the current family members—is a painful one. I've participated in discussions with software developers at companies that are struggling with the problems that exist just between the PC*jr* and the PC (which are not major) and the level of pain already is clear. System developers have begun to pick and choose among the options currently available; for example, programs are already appearing that do not support versions of DOS prior to 2.0. Others are incompatible with the older ROM BIOS chips. What's a body to do?

IBM should announce, now, a set of "core" standards. Let us know now that every future machine (say, for five years) will include support (even if it is at extra cost) for 360KB diskettes; for 16-color, medium-resolution graphics; for phantom memory where the monochrome display adapter usually has the real thing. Let memory between 640KB and 1MB remain forever reserved. Let the DOS 3.0 file system be emulated in any new environment.

I don't want us to accumulate a bag full of modes, configurations, and options so large that it becomes virtually unmanagable. I do want us to have at least one operational mode upon which we can depend, and depend for some time to come.

ABOUT OUR NEW LOOK

By now, many of you will have noticed changes to *PC Tech Journal*. As with all things, some of the changes may be to your liking and some may not. I want to let you know why we have made these changes and why we believe they represent improvements.

The most obvious change is in the size of our type, which is one point (one-sixtieth of an inch) smaller than it was before. You have often complimented us on the typeface, remarking mainly about its readability, so the change was a difficult one for us to make. Our art director, Nancy Lepow, and the art department were careful, though: in order to preserve clarity and readability, a concurrent, though subtle, change was made in the type style.

What makes the change so important, both for us and for you, is the additional editorial space we pick up in the same number of pages. *PC Tech Journal* articles tend to be longer and more complex than those in other magazines; the penalty paid was in not being able to run the number of articles we wished. Now we will be able to bring you at least one and often two more articles each issue, something we trust will make the magazine an even better value for you.

On a more sensitive note, we continue to receive comments about our printed listings. If you guessed that the complaints have to do with small type and fuzzy printing, you are right on the mark. We empathize with the bleary-eyed midnight reader struggling with a tilde that looks like a dead spider. I wish I could tell you now that we have solved the problem, but we are still working on it. We have developed a number of approaches and are experimenting to find one that gives the best quality of reproduction. Our ultimate goal: compact, typeset listings, with a crisp, readable appearance.

You will also begin to notice a greater consistency in the design and layout of our articles, as well as an increased use of color. A great deal of thought has gone into these changes, and the result is a magazine that is more attractive and easier on the eyes.

You will discover, I'm sure, many of the other changes we have made. We at *PC Tech Journal* hope that our efforts continue to meet with your approval and, as always, we welcome your comments.

AUTHORS, AUTHORS

In the short and interesting life of *PC Tech Journal*, we have received literally hundreds of unsolicited manuscripts. While it goes against the grain to say stop sending them, we do indeed have to say stop—with qualifications.

Please do not stop thinking about articles for *PC Tech Journal*. We rely on the collective expertise of a host of working professionals close to their disciplines to suggest articles and otherwise contribute to the ferment of ideas for the magazine. Only through a network of authors can we deliver our painstaking, detailed coverage of the IBM family of personal computers.

So if we don't want articles, what do we want? The answer is very simple: query letters. We expect them, get them, and use them productively. Queries save authors from spending time on manuscripts that are never used, and they enable us to read and assess ideas quickly. And I might add that a telephone call is no substitute for a letter: we cannot pass it around the office for all our editors to see and digest. Please don't phone.

Good query letters state the thesis of a proposed article and explain, most often in an outline, how the article would be developed and why you are the right person to write it. Letters should also include your background: current position, area of expertise, books or articles published, and current interests.

When we receive query letters from authors, we evaluate them according to the following criteria:

Background. Does your background relate to the topic that is being proposed? Do you have the technical background to tackle the subject matter? What is your formal training and experience?

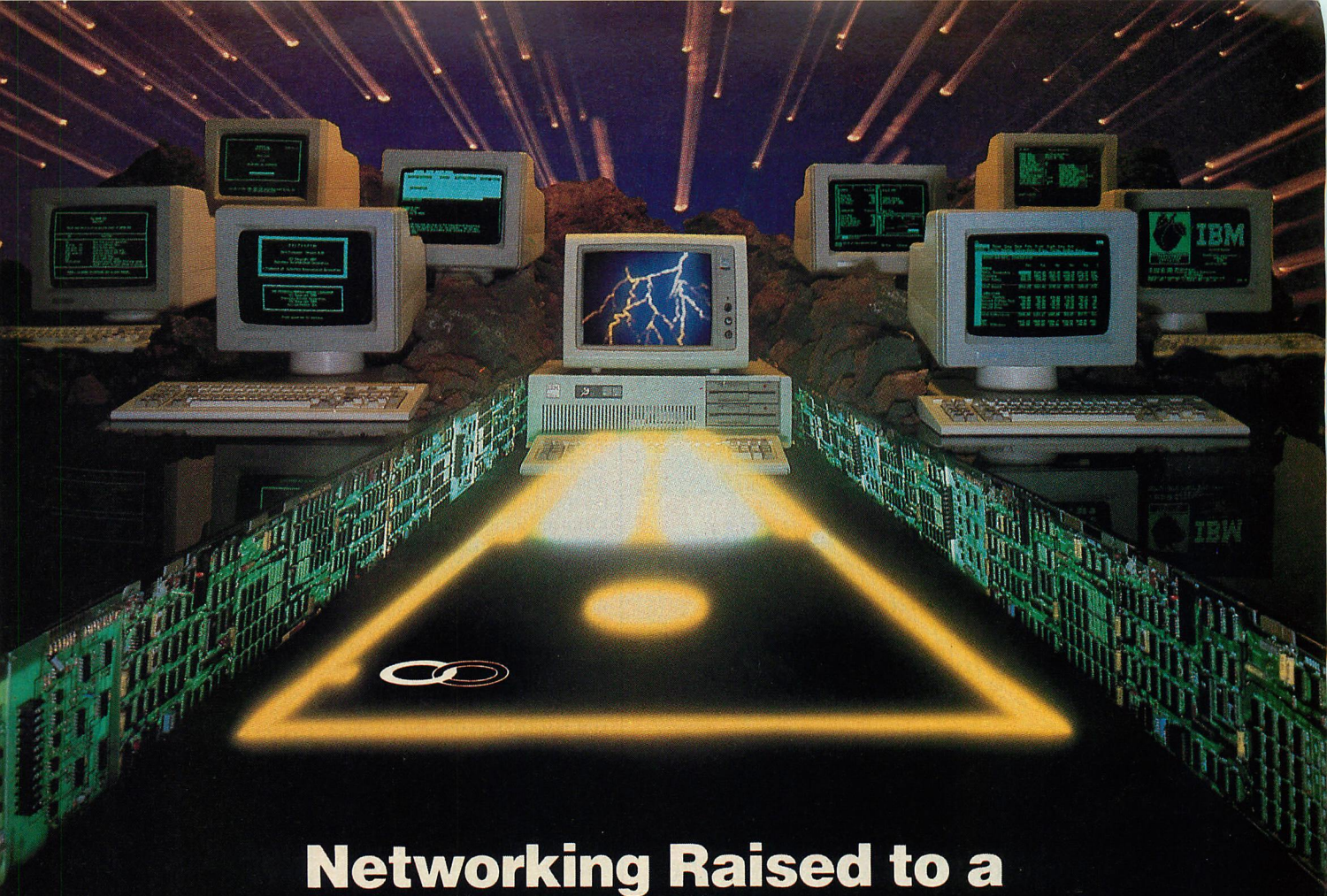
Writing ability. How well does your query letter express your interests? What can we learn about your style through your other published works?

Interests. Do your interests relate to our editorial needs?

Possible assignments. To which areas might you best contribute: overview articles, reviews, hardware or software or both, databases, UNIX, mainframes, documentation, Tech Notebooks, book reviews, or other departments?

A positive response from us to a query letter amounts to a telephone call from one of our editors to discuss the idea further. What often happens is that parts of the idea or approach we will like but other parts we will want to redirect. On the phone we meet a mutually acceptable plan, and then move on to a discussion of scheduling.

I encourage you to share your ideas for articles with us. It is my hope that we will continue to build on the solid base of expertise already developed by our contributors. The process of becoming a *PC Tech Journal* author starts with sharing ideas, continues with the hard work of putting words on paper, and ends with rewards for the magazine, its readers and authors.



Networking Raised to a Greater Power

Advanced Technology. With it, IBM tripled the speed of the PC and increased its memory capacity five-fold. Nowhere is this increase in computing power more important than in networking situations. If the AT's technological advances have prompted you to look into a multi-user network, you owe it to yourself to take a closer look at MultiLink Advanced™ . . . a unique multi-tasking, multi-user networking system that runs programs under PC-DOS 3.0.

Eight Workstations for the Price of an AT. MultiLink Advanced™ represents the next generation in networking systems for IBM microcomputers. The system enables terminals, connected to a single AT, to emulate IBM-PC's having up to 480K of RAM (The PC-Shadow™ terminal, shown above, even has a PC look-alike, as well as work-alike, keyboard and display).

This means that instead of spending $3,000 per workstation for a PC with a Kilobuck "Network Interface Board," you can use inexpensive terminals . . . eight of which cost less than an IBM AT. Even if you need only one workstation connected to your AT, you'll realize significant savings.

MultiLink Advanced™ . . . Instant Access to All of Your Resources. Central to most multi-user situations is the need to coordinate a variety of printers. With what's been described by *PC-Tech Journal* as ". . . by far, the best print spooler for the IBM PC," MultiLink Advanced™ gives users the option to print either at their workstations, or at a central location. In addition, programs and files can be shared by multiple users locally or through use of a modem. Just think of it . . . having remote access to an AT with a lightweight terminal/modem.

Although designed to take advantage of the AT, MultiLink Advanced™ runs on all versions of PC-DOS, except 1.0, and certain implementations of MS-DOS. A wide range of leading programs are supported which include WordStar, dBASE III, Multimate, and Lotus 1-2-3.

Get the Advanced Story Today. Call The Software Link Today for complete details and the dealer nearest you. MultiLink Advanced™ is immediately available at the suggested retail price of $495 and comes with a money-back guarantee. VISA, MC, AMEX accepted.

MultiLink™
ADVANCED

THE SOFTWARE LINK, INC.

8601 Dunwoody Place, Suite 336, Atlanta, GA 30338 Telex 4996147 SWLINK

CALL: 404/998-0700

Dealer Inquiries Invited

IBM, PC, AT, & PC-DOS are trademarks of IBM Corp. MS-DOS, WordStar, dBASE III, Lotus 1-2-3, and Multimate are trademarks of Microsoft Corp., MicroPro, Ashton-Tate, Lotus Development Corp., & Multimate International, respectively.

MultiLink Advanced™ is a trademark of The Software Link, Inc.

CIRCLE NO. 192 ON READER SERVICE CARD

Introducing

a complete internal half-high 60 megabyte tape backup system for your IBM® PC, XT, or AT for only $995

Complete with tape drive, controller, and software

And it works easily with all IBM compatible operating systems. Simply tell your tape drive you want to backup or restore the entire hard disk, or any file, or backup or restore by subdirectory, date, or everything since you last backed up.

Now you won't have to wait for the difficult to find hard-disk version of the IBM AT (model 99). You can buy the floppy disk IBM AT (model 68), add our hard disk and tape drive system for about what you'd pay for the AT hard disk upgrade alone. It's almost like getting the tape drive free.

Let's face it, we've all heard the horror stories of people who've lost data on their hard disk. True, it doesn't happen often, but then disaster seldom does. With the amount of data you can put on a hard disk these days, no one in business can afford even a small disaster.

When did you last backup your hard disk?

Oh, you did it once with floppies and it was so time consuming that now you've convinced yourself nothing will go wrong? In other words, it can't happen to you. And besides, at the prices they're asking for tape backup—$2,000 and up—you're willing to take a chance. You've seen some tape drives for less, but you have to buy an expensive hard disk to go with it, and you've already got a good hard disk. Where can you turn for relief?

IBM Compatible tape drive system complete for $995

The Express Systems™ tape drive comes complete—half-high tape drive, controller, and software—for only $995. It's absolutely IBM compatible—all 60 megabytes of it.

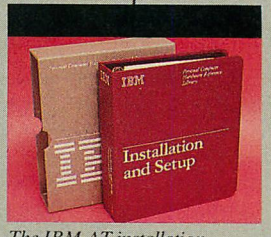
You can use your tape drive in the event your hard disk fails. And if you have to replace your hard disk, the tape's ability to read bad sectors will let you replace your hard disk with another even if the new one is not error-free. The tape requires very low power, too.

And it doesn't poke along. It reads and writes at 90 inches per second (ips) and transfers data at up to 3.75 megabytes per minute in the streaming mode. You don't have to be a rocket scientist to figure that you can perform an image backup of a 20 megabyte hard disk in about 5 minutes. But practically speaking, once you back up your hard disk completely for the first time, you never need to do more than invoke the archive command—that convenient command that tells your new tape drive to back up everything since you last backed up. If you backup as often as you should, your Express Systems tape drive will finish the job virtually in seconds.

The Express Systems software has additional benefits, like enabling you to use PC DOS terminology such as "*", "*.*", and "?". It also has a built-in reformatter, built-in verification (to make sure you transferred what you thought you did), and it's prompt driven, which means it asks you exactly what you want to do.

Easy to install

Before you get intimidated about installing our tape drive internally, you should understand that IBM doesn't think it's too difficult. They're selling IBM PC ATs with instructions on how to add additional hard disks in the *Installation and Setup* manual that comes with the AT.



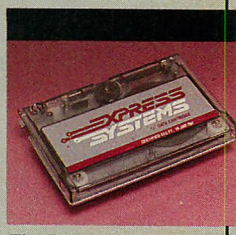
The IBM AT installation manual shows how easily you can install internal storage drives yourself.

Our instructions for installing your new Express Systems tape drive follow IBM's clear, simple instructions.

We even provide the tape cartridge

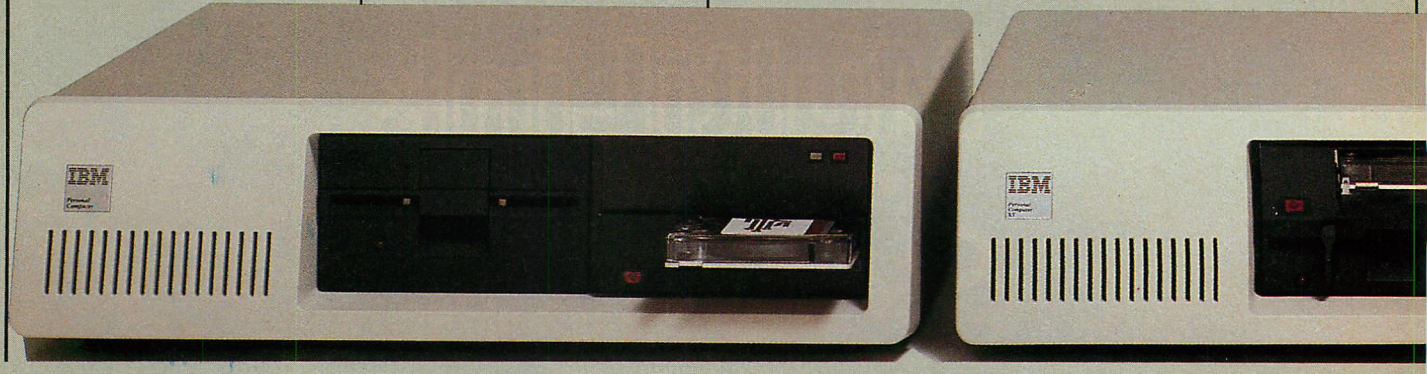
Most people don't realize that the tape cartridge contains most of the critical mechanisms to insure data integrity. In order to be sure that you get the best insurance for your data (after all, isn't that why you're buying it) we encourage you to use Express Systems' specially tested tape cartridges. We're not going to kid you and tell you others won't work, but here's what's special about Express Systems' tape cartridges.

First, they are tested down four separate tracks from end-to-end, not just down the center of the first 150 feet, like some others do. We use three screws to hold the cover on instead of four. This simple triangular arrangement keeps the baseplate flat, just like three legs work better than four to make a table steady. Since all tape drives reference everything to the baseplate, this alignment is critical. We also use special rollers to dissipate posible static electricity buildup—something that can ruin your whole day.



The Express Systems tape drives come with Express Certified™ 555 or 600 ¼-inch tape cartridges with quadruple end-to-end testing for extra insurance of your data.

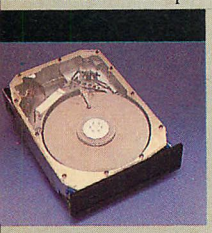
And finally, we will sell you tape cartridges in boxes of three instead of the usual five. So, you get higher quality with a smaller quantity commitment. And we compound the savings with a lower per unit price, just $35.00 instead of the usual $45.00 most retailers charge.



Need a hard disk?

Depending on whether you have an IBM PC, XT, or AT you may want additional hard disk storage. We have those too. We offer 10, 21 and 31 megabytes of formatted hard disk storage.

For the most part, our drives are made with plated media, which means there is less chance to damage them. (Let's face it, the oxide that most disks come with is nothing more than rust.) We then test the drives, pre-format them, and install DOS 3.0 so that you're ready to begin transferring files. We even include DOS 3.0 documentation.

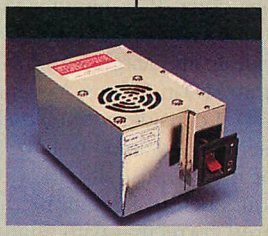


Express Systems offers 10, 21, and 31 megabytes of formatted storage in the half-high form so you have extra space for other storage options.

And they're 100 percent IBM compatible. The controller we send you for the XT is an upgraded version of the XT controller from the same company that makes the XT controller. In fact, the Express Systems controller is an improved controller which requires less power so that it is more reliable than any other standard controller.

We provide the power too.

If you want to upgrade your IBM PC, there just isn't any way around upgrading your power supply—if you want to have true XT or better capability. Some companies say that their hard disks don't require any increase in power—and they might be right. But don't add anything to your slots, because the minute you do, you'll need more power. That's the bad news.



Our 130 watt power will convert your PC to XT standards; and it's only $99.50 with any tape drive or hard disk order.

The good news is that our power supplies are inexpensive. How's $99.50 for an XT power supply? We mean a full 130 watts of power. The other good news is that it's held in by only 4 screws. Express Systems' power supplies can be changed in 20 minutes, a small price in time for the peace of mind to convert your PC to an XT-capable machine and avoid the unsightly "wart-like" power supply add-ons that some companies insist you paste on the back of your PC.

Express Systems Upgrade Kits

(Includes controller, software, and cable where appropriate)

IBM AT (model 99) to AT ExPlus™
1 half-high tape drive system — **$995**

IBM XT to XT ExPlus
1 half-high floppy and 1 half-high tape drive system — **$1095**

IBM AT (model 68) to AT ExPlus

21 Megabyte upgrade — **$1895**
1 half-high 21 megabyte hard disk with half-high tape drive system

31 Megabyte upgrade — **$2195**
1 half-high 31 megabyte hard disk with half-high tape drive system

IBM PC to XT ExPlus
2 half-high floppies, 1 half-high 10 megabyte hard disk with controller, 1 half-high tape drive system and 130 watt power supply — **$2095**

But from a mail order house?

We get tired of the snide remarks some people make about mail order houses. The comments are usually spread by distributors and retailers who are getting cut out of 15 and 35 percent margins, respectively. If we went through distribution—you'd have the privilege of paying for large glass windows, rugs, salesmen, etc.—but we'd also be selling this tape drive for $1495.

We're not criticizing distributors and retailers. They perform a valuable service. But you don't need them if you know what you want. And you can certainly install it yourself. IBM has proved it with their instructions for self-installation that come with the new IBM PC AT.

And speaking of IBM, the next time you hear anyone criticize mail order as a way to buy computer equipment, remind them that IBM is now in the mail order business.

Warranty

We offer you a one year warranty on our hard disks—the same as IBM on the AT and 90 days on the tape drives. (It's all the manufacturer gives us.) If anything goes wrong with your tape or disk drive or hard disk, send it back in the box it came in. However, we have found that we can usually solve the problem over the phone. So call first for a return authorization number because we can't accept any returns without it.

Immediate delivery

We have four types of delivery: *Next Flight Out*℠ if you need it immediately; *Next Day Express*℠ and *Day After Tomorrow*℠ if you can wait a day or two; and our normal delivery—which we pay—if you can wait a few days.

Express Systems Product Line

Internal Tape System
Half-high tape drive, controller, & software — **$995**

Tape Subsystem
Tape system, controller, power, supply, cable, and chassis — **$1195**

Internal Tape & H.D. System
Tape & hard disk internal systems come complete with tape controller and hard disk controller, software, and cables. Hard disks are formatted and tested with DOS 3.0.

Tape drive & 10 MB H.D. — **$1695**
Tape drive & 21 MB H.D. — **$1895**
Tape drive & 31 MB H.D. — **$2195**

Hard disks Kits
(includes Controller and cable)
10 megabyte hard disk* — **$695**
21 megabyte hard disk* — **$995**
31 megabyte hard disk* — **$1395**
**Subtract $195 for IBM AT which does not require hard disk controller*

Controllers
Hard disk controller — **$195**

Floppy Disk Drive
Half high floppy disk — **$129**

Power Supply
130 Watt XT replacement Power Supply — **$99.50**
(Price valid only with purchase of tape drive or hard disk. Otherwise, $149.95)

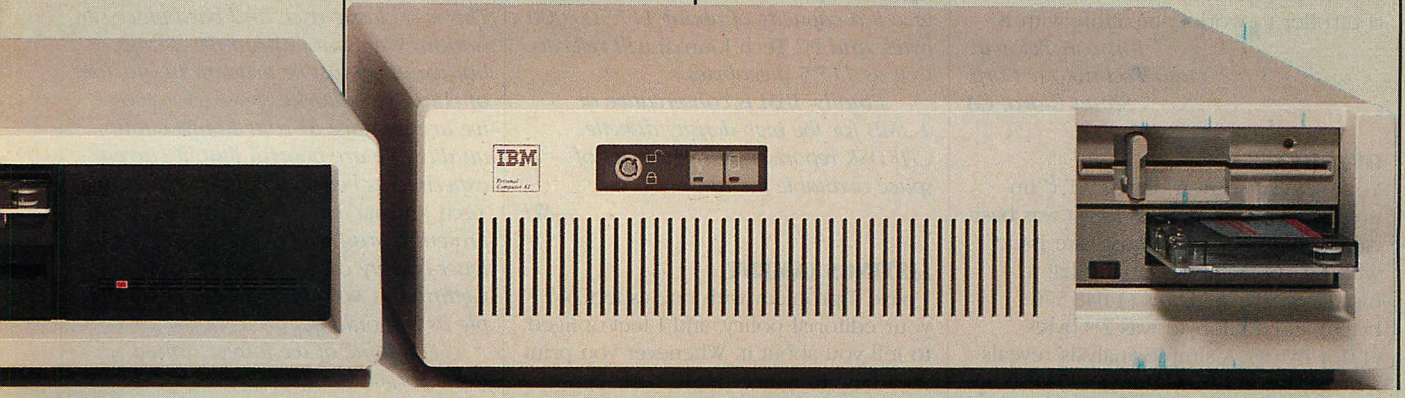
Tape Cartridges
Express Certified 555™ — **$35.00**
(555 feet of specially tested 310 Hci tape)
Express Certified 600™ — **$35.00**
(600 feet of specially tested 550 Hci tape)

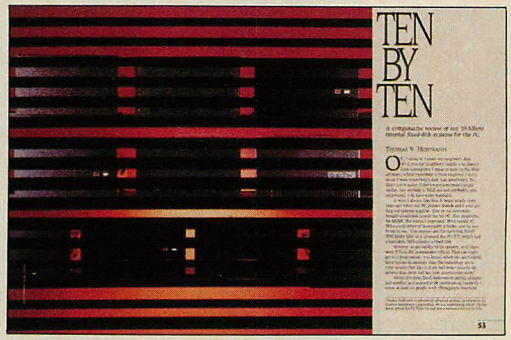
How to order

Pick up the telephone and call 1-800-341-7549, to order. We accept Master Card, VISA, American Express and Diners Club. Or send a cashier's check or money order (We'll take a check, but you'll have to wait for it to clear) and tell us if you want one of our recommended configurations or you want to mix and match yourself. Corporations with a DUNS number may send purchase orders for quantities over five.

EXPRESS SYSTEMS

Call Toll Free 1-800-341-7549 Ext. 100
In Illinois call (312) 882-7733 Ext. 100
Express Systems, Inc., 1254 Remington Rd., Schaumburg, IL 60195





ASKING FOR ASCII

The "Significant Figures" series by Robert Gray (October 1984, p. 54; November 1984, p. 173) is excellent. The articles are well written, understandable, and the programs are a needed addition to my programmer's toolbox. I do have one suggestion, however. How about another article that deals with ASCII-to-floating-point conversion. This would be very useful since BASIC stores numbers on diskette in ASCII format. Specifically, algorithms and assembly language routines for the equivalent of BASIC's VAL and STR$ functions would complete the series nicely.

Alan J. Testani
Apalachin, NY

Thanks.

By now, of course, you will have seen the December issue and the final installment of the series on—you guessed it—ASCII conversion. —WF

DTC COMPATIBLE WITH PC/IX

Thanks for the interesting articles on hard-disk systems in the November issue ("Ten by Ten," Thomas B. Hoffmann, p. 52; and "Fixed-disk Benchmarks," William J. Hunt, p. 64). I was happy to see that DTC's PC Hard Disk Controller (5150BX) fared well in your tests. Since you noted that none of the "compatible" controllers worked with PC/IX, I would like to update you. After PC/IX was released, we modified our controller to work compatibly with it.

Jon von Zelowitz
Data Technology Corp.
Santa Clara, CA

OVERBYTES

Your special section on the AT ("Impressive AT," December 1984) was basically very good. But I must take issue with one point. I was taught that a megabyte was 2^{20} bytes (1,048,576 bytes) and a kilobyte was 2^{10} bytes (1,024 bytes). A simple analysis reveals a megabyte is 1,024 kilobytes. These figures comply with the *American National Dictionary of Information Processing*. In numerous places in the articles in that section, there appears to be no consensus on the definition for megabyte. To be sure, you are not alone. Even IBM, in its product literature on the AT, lists the capacity of the high-capacity diskette at 1.2 MB (1,228,000 bytes). In reality, it is 1.17 MB (rounded). This is a minor point at lower capacities, but is progressively greater (in real terms) at larger capacities.

John H. Cato, Jr.
Nicholls, GA

No matter what the American National Dictionary of Information Processing *says, the prefix kilo- means thousands (as in 10^3), mega- means millions (10^6), and giga- means billions (10^9). We in the industry are to blame for bastardizing these meanings for our purposes.*

The notion of a kilobyte (1,024) is only useful when dealing with main memory; this is an artifact of the way in which memory cells are physically organized. There are some advantages to the scheme: both hexadecimal and octal notations of memory addresses are convenient. PC Tech Journal *will continue to use this definition of kilobyte (KB) when discussing main memory.*

For mass storage (or anything else being measured for size), the actual capacity is the important matter. AT drive type 9 has physical characteristics that give it a capacity of about 117,500,000 bytes, and PC Tech Journal *will refer to that as 117.5 megabytes.*

Finally, IBM is conservative at 1.2MB for the high-density diskette. CHKDSK reports 1,213,952 bytes of space available for use.

—WF

LISTINGS LAMENT

I have a problem with one aspect of your editorial policy, and I feel obliged to tell you about it. Whenever you print an article in which BASIC code is presented for the reader to copy, you can really botch it up. Not only are there errors in the code, which result from someone trying to fit copy to the narrow printed column, but they are so small that they are hard to copy.

If you are going to continue to print BASIC code, why not do it right. Use actual listings rather than typeset reproductions. Do not allow editors to break up the lines of code to fit a narrow column. If you want to make the code fit the column, then do it on the computer where you can check to see if it still works. Use an output printer that uses a slashed zero to distinguish the zero from the letter *O*. And, finally, print the code large enough so that those of us over 50 can tell a *B* from an *8* without using a magnifying glass. And leave off the halftone horizontal bands; they don't do a thing for the readability.

Today I tried to key in the BASIC code from "Aargh, I Didn't Save My Program" (Dan Rollins, November 1984, p. 134). I considered the article to be very useful and wanted to have the utility for my own use. It took a couple of hours before I got all the hex code entered correctly—with a little luck and the use of a magnifying glass. You have a great thing going in publishing useful utilities and small programs, so why don't you do it right.

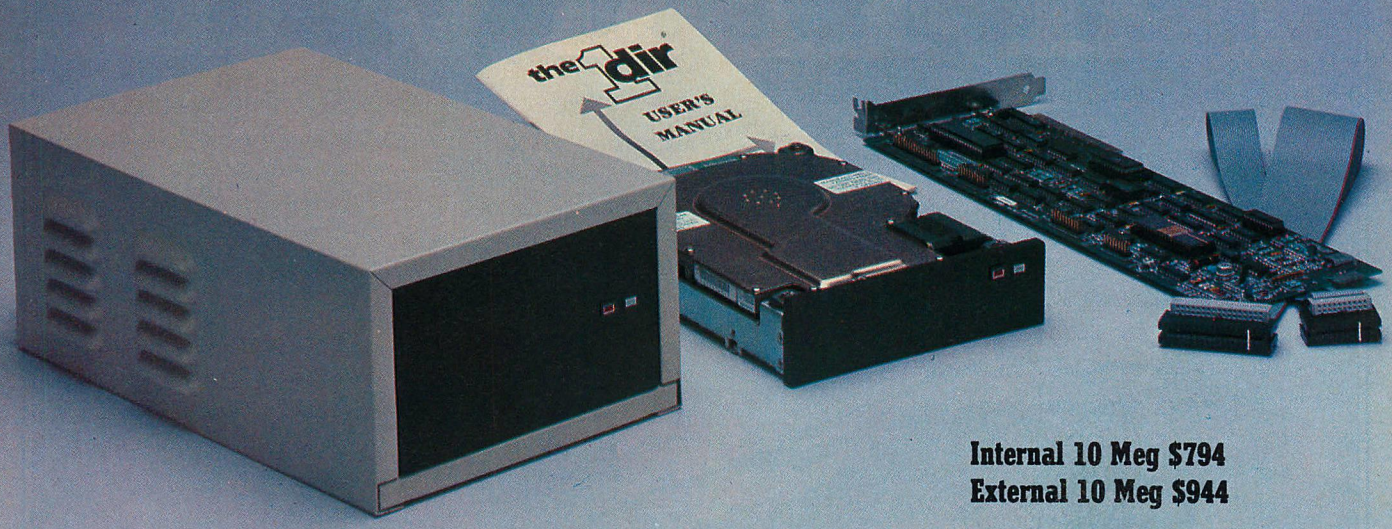
John Mitchell
Webster, NY

This is a classic rock and hard place situation. We want to print the listings, but we also must be prudent in our use of valuable editorial space. However, we are sensitive to your points about quality. We are hopeful that, if current experiments bear fruit, listings in PC Tech Journal *will be typeset (they are at present reproduced directly from actual letter-quality copies of the programs). Getting this accomplished is not as simple as it should be.*

For some of the longer listings, don't forget our listing diskette.

—WF

The Hard Disk With The Software Shell

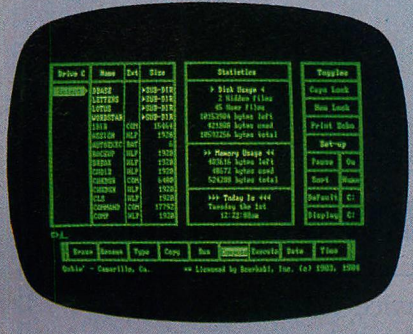


Internal 10 Meg $794
External 10 Meg $944

Why pay more for 10 or 20 Meg drives than you have to? Our 10 Meg internal hard disk subsystem is priced at $794, with the 20 Meg model going for $1088. Our external 10 Meg goes for $944, 20 Meg for $1238.

Our drives are fully compatible with any IBM PC or PC-compatible with 64K RAM and PC-DOS 2.0 or later.* Qubie' drives boot directly from the hard disk. You can power up the PC and load the system directly, without using any floppy disks. No software patches or drivers to install.

Using the same amount of power as a floppy drive, the Qubie' hard disk uses less energy than other aftermarket drives.



The drives come complete with 1dir software. 1dir's commands are in English, eliminating the need to type in DOS commands, and are all selected by using cursor control keys. 1dir even explains commands with HELP screens that give you on-line advice when you need it.

Qubie' drives are made of special plated recording media. They withstand the vibration and movement that has damaged hard disks in the past. In fact, Qubie' drives have been selected by several computer makers for use in their portable computers.

Good service starts with answering your questions before and after you buy. It continues with same or next day shipment of your order. Since we only sell a few selected products, we have the information and inventory to help you fast.

We perform repairs in our own service department within 48 hours, should you ever need service during the one year warranty period.

Our price is the whole price. All prices include UPS surface charges and insurance. In a hurry? Two day UPS air service is just $12.

Corporations, dealers and institutions, call for volume purchase price information.

*Call for information.

CIRCLE NO. 224 ON READER SERVICE CARD

No Risk Guarantee
If you are not completely satisfied with your purchase, you may return it within 30 days for a full refund, including the cost to send it back. If you can get any of our competitors to give you the same guarantee, buy both and return the one you don't like.

Order Today, Shipped Tomorrow!
For fastest delivery, send cashiers check, money order, or order by credit card. Personal checks, allow 18 days to clear. California residents, add 6% sales tax.
Hours: Mon.-Fri. 8:00 a.m.-6:00 p.m. PST
Sat. 9:00 a.m.-1:00 p.m. PST

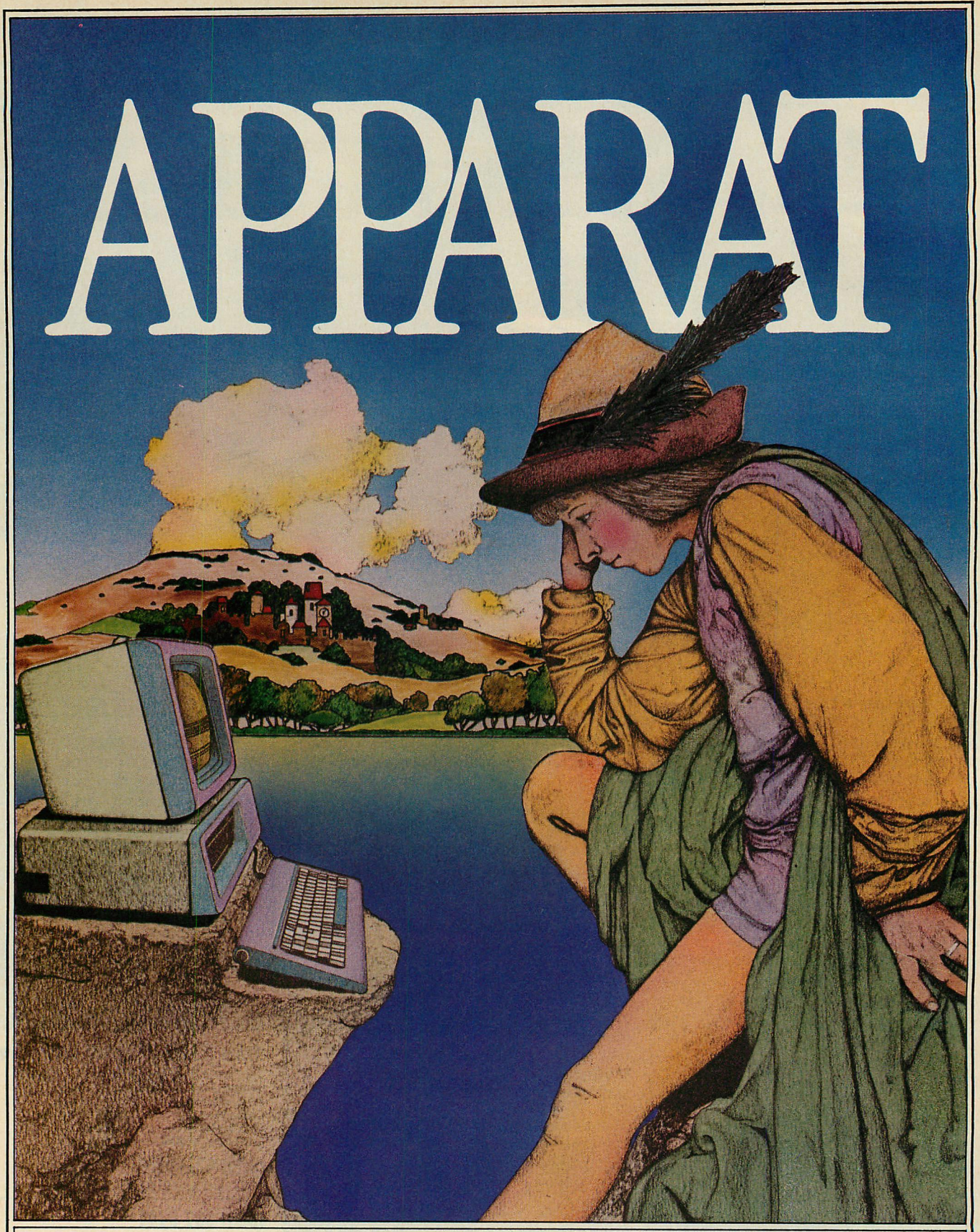
(800) 821-4479
Toll Free Outside California

(805) 987-9741
Inside California

QUBIE'

4809 Calle Alto
Camarillo, CA 93010

London (01) 223-4569
Paris (01) 321-5316
Sydney (02) 579-3322



APPARAT TAKES YOUR PC TO NEW REALMS.

4401 South Tamarac Parkway, Denver, CO 80237—Customer Service 303/741-1778
Apparat retail outlets—DENVER: 4401 S. Tamarac Pkwy. 303/771-2032 • 5224 W. Warren Ave. 303/985-1217
CHICAGO: 1844 S. Arlington Heights Rd. 312/640-0322

CIRCLE NO. 106 ON READER SERVICE CARD

LETTERS

HARD DISKS, FIXED FELINES

I find it rather strange that IBM should call its hard-disk system *fixed* because the medium cannot be removed (see "The Hard (or is it Fixed?) Disk Revolution," Will Fastie, November 1984, p. 5). I call my cat, the Itsy Bitsy Mouser, *fixed* because his drive has been removed.

Harry Gross
Ottawa, Ontario

INTERFACE CLARIFICATION

I can sympathize with Dennis During's confusion about the interface between MS-Pascal Version 3.2 and the 8087 math coprocessor in the IBM PC and XT ("Pascal Opinions," Letters, October 1984, p. 12). I had an identical run-in with the Microsoft documentation. However, there are two excellent references that clarified the issue to me: "Interrupts and the IBM PC" by Chris Dunford (*PC Tech Journal*, November/December 1983, p. 173); and *Interfacing to the IBM PC* by Lewis Eggebrecht.

Microsoft's documentation is not wrong. Neither are members of the support staff when they state that no patches are required, although they appear to lack a clear understanding of the IBM PC interrupt scheme. The confusion seems to arise from Intel's interrupt numbering scheme on its 8086/8088-based systems (the IBM PC is one). There are two numbering schemes: the microprocessor's and the 8259 controller chip's. The former are usually denoted by simple numbers (0, 1, 2, etc.), while the latter are denoted by numbers with the prefix "IRQ." The microprocessor on the PC has five interrupts connected directly into it—divide error, single step, NMI, INT instruction, and INTO instruction—numbered in that order from 0 to 4. Then there are three more: two reserved and one for the BIOS print screen. *Then* come the 8259 interrupts. An interrupt channel on the 8259 presents an interrupt of level (channel number + 8) to the 8088. Thus, an interrupt on the 8259's channel 0 (timer) corresponds to an 8088 interrupt level 8. Thus, when Mr. During states that the Microsoft Mouse uses interrupt 2, what it means is that it uses 8259 channel 2, i.e., interrupt 10. There should be no problems with the Mouse conflicting with the 8087, unless some other device in his system is already using this channel.

Thus, the NMI is, in fact, interrupt number 2, and it is not fed through the 8259. According to Eggebrecht, interrupts from the 8087 are ORed with the system board parity check interrupt (the one that causes you to lose six pages of carefully typed document) and the I/O channel check interrupt, before being fed directly to the 8088's NMI port. The 8259 takes care only of interrupts numbered 8 through 16.

I am shocked that the staff at Microsoft was unaware of this. After all, they designed the compiler to support this particular architecture.

There is one more extremely important fact that Mr. During should be aware of. The instructions given for installing the 8087 in IBM's documentation are wrong. Switch number 2 on switch block 1 should be set to OFF when the 8087 is installed. This enables the whole interrupt scheme described above. If this switch is on, as IBM suggests, 8087 interrupts will not be reported to the 8088. As a result, if an MS-Pascal program causes a real number overflow or divide by 0, the computer will go west and require a reboot.

I hope this resolves Mr. During's problem. The upshot of all this is that, yes the MS-Pascal 8087 interrupt system supports the IBM PC architecture, and no patching is required. However, don't forget to set that switch to OFF.

Kuryan Thomas
Blacksburg, VA

C-NOTES

There was a major omission from the article "BASIC to C" (Ernest Tello, October 1984, p. 117). Programmers who are not aware of it may find themselves in a large bind in the future.

The cause of the omission is probably due to the general confusion over the phrase *portable language*, as in "C is a portable language." People believe this means that source code written in C, the portable language, can be transported to other computers, operating environments, and compilers. Not true.

A language that enables the source application to be transported to other computers, operating environments, and compilers is a *standard machine independent* language. The prime example of such a language is ANS PL/1. Other languages, such as FORTRAN 77 and COBOL, are also standard machine independent languages.

Portable means that C itself can be transported to other computers and operating environments. This is possible because the language is large enough that the compiler can be written in C. The language is also sufficiently machine independent that it looks and works the same on different kinds of machines.

While this transportability is not unusual for languages like PL/1, C is dif-

NEW from BORLAND!

TURBO TOOLBOX & TURBO TUTOR

Offer extended by popular demand!

Get your Borland Holiday Pack by March 1st, 1985.

"TURBO is much better than the Pascal IBM sells."

Jerry Pournelle,
Byte, July 1984

"TURBO PASCAL appears to violate the laws of thermodynamics.

You won't find a comparable price/performance package anywhere. It is simply put, the best software deal to come along in a long time. If you have the slightest interest in Pascal . . . buy it."

Bruce Webster,
Softalk IBM: March 1984

BORLAND INTERNATIONAL GIFT PACK

ONLY **$99.95**

A SAVINGS OF $30!

What a gift for you and your friends! The extraordinary TURBO PASCAL compiler, together with the exciting new TURBO TOOLBOX and new TURBO TUTOR. All 3 manuals with disks for $99.95.

TURBO PASCAL Version 2.0 (reg. $49.95). The now classic program development environment still includes the FREE MICROCALC SPREAD SHEET. Commented source code on disk

- Optional 8087 support available for a small additional charge

NEW! TURBO TOOLBOX (reg. $49.95). A set of three fundamental utilities that work in conjunction with TURBO PASCAL. Includes:

- TURBO-ISAM FILES USING B+ TREES. Commented source code on disk
- QUIKSORT ON DISK. Commented source code on disk
- GINST (General Installation Program)

Provides those programs written in TURBO PASCAL with a terminal installation module just like TURBO'S!

- NOW INCLUDES FREE SAMPLE DATABASE . . . right on the disk! Just compile it, and it's ready to go to work for you. It's a great example of how to use TURBO TOOLBOX and, at the same time, it's a working piece of software you can use right away!

NEW! TURBO TUTOR (reg. $29.95). Teaches step by step how to use the TURBO PASCAL development environment—an ideal introduction for basic programmers. Commented source code for all program examples on disk.

30 DAY MONEY BACK GUARANTEE Available at your nearest software dealer.

For VISA and MASTERCARD order call toll free: **1-(800)-255-8008 1-(800)-742-1133**
(Lines open 24 hrs., 7 days a week) Dealer and Distributor inquiries welcome (408) 438-8400

CHOOSE ONE (please add $5.00 for handling and shipping U.S. orders)

___ All Three-Gift Pack	$ 99.95 + 5.00 **SPECIAL!**	___ Turbo Toolbox	$49.95 + 5.00
___ All Three & 8087	139.95 + 5.00 **SPECIAL!**	___ Turbo Tutor	29.95 + 5.00
___ Turbo Pascal 2.0	49.95 + 5.00	___ Turbo 8087	89.95 + 5.00

Check_____ Money Order_____ VISA_____ MasterCard_____

Card #: _____ Exp. date: _____ Shipped UPS

My system is: 8 bit ___ 16 bit ___

Operating System: CP/M 80 ___ CP/M 86 ___ MS DOS ___ PC DOS ___

Computer: _____ Disk Format: _____

Please be sure model number & format are correct.

NAME: _____

ADDRESS: _____

CITY/STATE/ZIP: _____

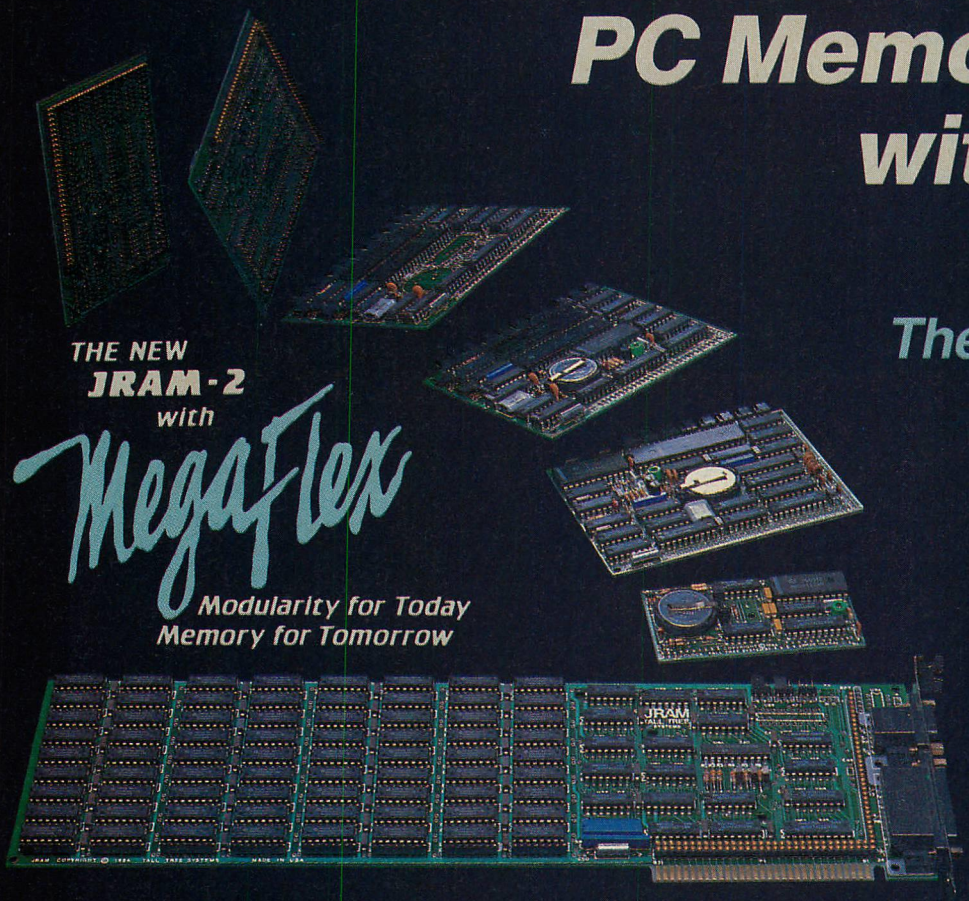
TELEPHONE: _____

California residents add 6% sales tax. Outside U.S.A. add $15.00 (if outside of U.S.A. payment must be by bank draft payable in the U.S. and in U.S. dollars). Sorry, no C.O.D. or Purchase Orders.

BORLAND INTERNATIONAL

4113 Scotts Valley Drive
Scotts Valley, California 95066
TELEX: 172373

Megaflex Your PC Memory Muscle with JRAM-2.



THE NEW **JRAM-2** with

Megaflex

Modularity for Today
Memory for Tomorrow

The first 2 megabyte multi-function memory board for IBM PC and all 100% compatibles...

Increase your computing power with Hardware and Software solutions for PC'S and compatibles...

- **EXPAND UP TO 2 MEGABYTES PER BOARD.** Add 256K chips one bank at a time as desired.
- **MODULAR I/O PORTS – EASY TO CHANGE AND UPGRADE.** The JRAM-2, with module, fits in a single slot.
- **FINEST CONNECTOR TECHNOLOGY – NO INTERNAL CABLES.** All add-on slots are easily accessible.
- **MOST POWERFUL SYSTEMS SOFTWARE – JETDRIVE AND JSPOOL.** Unparalleled power and increased computing speed.

PURCHASE ONLY THE FUNCTIONS YOU NEED.

CHANGE MODULES AS YOU WISH.

AVAILABLE NOW:

- JRAM-2 main memory board
- JRAM-2 with Clock-Calendar
- JRAM-2 with Clock-Calendar, Serial Port, and Parallel Port
- JRAM-2 with Clock-Calendar and two Serial Ports
- JRAM-2 with two Serial Ports
- JRAM-2 with diskette controller

All the memory capacity you'll ever need!

1032 Elwell Court,
Suite 124
Palo Alto, CA 94303
415/964-1980

TALL TREE SYSTEMS

CIRCLE NO. 191 ON READER SERVICE CARD

ferent because it is not really a high-level language. It is a relatively small language that primarily operates upon single bytes augmented by the capability of building up functions and TYPEDEFs. C is an assembly language that grows through its function library.

Since the function library is written in C, once the compiler is moved to the new environment, the library can be recompiled, making C fully available.

BASTOC depends upon a specialized function library that is not distributed as source. This means that a program that was translated through BASTOC is less transportable than the original BASIC program was.

If BASTOC were advertised as a restructuring tool and distributed with the source code for the function library, I would recommend it to other programmers. At it is, I find BASTOC to be marginally useful, and I cannot recommend it to other software developers.

Cory Hamasaki
Alexandria, VA

While there are some correct statements in Mr. Hamasaki's letter, there are some major misconceptions, and the main thrust of it is not entirely fair to JMI, the vendors of BASTOC, or me. The issue of C portability is a complex and controversial one, requiring an article by itself. I deliberately made no attempt to address it my article.

The "greater portability" to which I attributed part of C's increasing popularity is a result of the solidly established standard for the language set by Kernighan and Ritchie's The C Programming Language, *and by the C standard I/O library accessed by the stdio.h header file. I have never heard it suggested that this means that any and every program written in C is automatically portable. Most beginners' texts on the language contain suggestions and techniques on how C can be used to write portable programs or program modules.*

It would seem to me to be an obvious contradiction to say that C compilers written in C are transportable to other computers and operating environments but other programs are not. Apart from the obvious difficulty of already needing a C compiler to compile the compiler on a target machine, it is also clear that it would be much easier to write many other types of portable C programs than a portable compiler.

It is nevertheless true that as languages go, C is quite small and requires additional libraries for many types of applications and that those libraries must be either available in portable C

source or in object modules on the target machine for such applications to be truly portable. I wrote about some of the major limitations of BASTOC in the article that should make developers think carefully before deciding that it is the right tool to use. The portability of its library, however, does not appear to be one of these limitations. On the contrary, JMI seems determined to port BASTOC to every imaginable environment. Of course, the library for each target has to be purchased separately.

—Ernest Tello

PORTABLE SUPPORT

Poor Peter Hanson ("Portable's Complaint," Letters, August 1984, p. 14). Traded his PC Portable for a PC/XT because of limited expansion just to try to run IBM's limited cluster network (see "Clusters of Wrath," Susan Glinert-Cole, August 1984, p. 64). He missed the main feature of the PC entirely: the aftermarket. Too bad that his dealer didn't tell him that the Portable (like the XT) has jumpers and socket to put 640KB on the system board using the new 256KB chips for about $350, or that several companies offer 10- to 20MB half-height drives and controllers (that fit in J2) from $900 to $1,300 (i.e., Qubié), or that the Tecmar Bosun half-card (that fits in J6 with parallel connector in J8 slot) has serial/parallel ports and battery clock—all for about $150, or, in short, a number of things.

For the $4,395 he spent on the XT system unit plus the $500 or so on the display adapter, parallel port, and display, he could have had a 640KB, 20MB machine with clock and ports *plus* a 1,200-baud Ventel half-card modem (in J4 or J5) or a copy of Framework or Symphony and an 8087 with slots left for a game and mouse controller (in J4/5 and J7), and the possibility of an internal RGB monitor later.

My portable had four banks of 64KB RAM. Quoted RAM cost above is based on purchasing two banks of 256KB chips and selling two banks of 64KB chips for about $100. If IBM is now using one bank of 256KB chips, the cost of one bank of 256KB and two of 64KB chips would be similar. The only card IBM supports for J8 on the XT is the RS-232 (the card has to return a CRD SELTD signal) that will not physically fit in the Portable. However, I suspect that if the drives were removed, the IBM RS-232 card would actually function and that J8 is fully operational, but I don't have the card to try it.

Maurice Riggins
(no city given) NY

PCs IN LOCAL GOVERNMENT

I look forward to receiving your publication. I find it helpful in keeping track of the IBM PC environment. I hope that your other readers can help me. I am chairperson of the Computer Study Committee for our town. I am interested in packaged systems that run on an IBM PC, XT, or compatible that can track property assessments, tax bills, etc., in accordance to Massachusetts requirements. In addition, I am interested in packages for school budgets, town budgets, water department, fire and police, etc. The readers can send the information to my office: Rebello Associates, 129 Main Street, West Harwich, MA 02671. Thanks for your help.

Robert E. Rebello
West Harwich, MA

DVORAK WARNINGS

With all the interest in the Dvorak keyboard layout, many people are popping the keycaps and rearranging them on their IBM PCs. They should first heed several warnings:

- Do not remove the space bar or you will never get it back on again properly unless you have four hands and biblical patience.
- Before you buy a keyboard, be sure all the keys are shaped identically. If the keys are sculpted, you will not be able to rearrange them physically.
- If you want to switch to Dvorak, you must quit QWERTY cold turkey. Switching back and forth is not only painful, your typing speed drops under both systems. If you are temporarily forced back to QWERTY, don't touch-type—use four fingers at most.
- It is almost impossible not to touch-type with Dvorak. The layout subtly seduces you into it. If you value your reputation as a machine gun two-finger typist, stick with QWERTY.

Roedy Green
Vancouver, B.C.

SHARING OPINIONS

The article "Shared Files" by William Colsher (November 1984, p. 99) was a good review of the problems that must be solved while putting up an EtherShare networked application. But the statement on page 106 that "all shared files that can change size must be preallocated" is not precisely correct.

There is a safe method by which a file can be dynamically extended by one application, then shared across the network, without that file being preallocated. It involves using the PC DOS Disk Reset function call in the "reading" node, which apparently forces the local

Discover Solutions With Vision

Dealers are finding Pacific Datanet's Vision™ hard disk sub-systems are today's personal computer users' choice for mass storage sub-systems.

The Vision Series is more than illusionary—read on!

Capacity *A variety of sub-systems—12/10Mb, 19/17Mb, 26/21Mb, 40/32Mb, 50/43Mb, 70/62Mb, 105/92Mb, and 140/125Mb.*

Software *Menu-driven software makes installation simple with the IBM PC, XT, or IBM compatible.*

Complete Sub-System *Vision Series are complete. Sub-systems include a 5¼" Winchester, our controller, host adapter, heavy duty international power supply, software and cables.*

Fast Tape Backup *5¼", ¼" tape streamer backs up at a consistent 5Mb per minute, using standard 90 ips, ¼" tapes.*

Error Correction *Sub-systems provide 8 bit error correction.*

Reliability *Stringent quality control assures reliable operation.*

Price *Prices start at $2,195 for the hard disk sub-system, and $1,995 for the streaming tape backup.*

Dealer Inquiries Invited

PACIFIC DATANET, LTD.
4701 Patrick Henry Drive, Bldg. 9
Santa Clara, CA 95054
(408) 980-0693 Telex: 759341

Pacific Datanet

EtherNet driver to fetch a fresh copy of the FAT (File Allocation Table) for the network volume from the server.

- The EtherShare volume where the file is to reside must have the attribute "shared."
- Node 1 opens the file for read/write, seeks to end of file, adds some records, and closes the file again.
- Node 2 closes all currently open files.
- Node 2 executes DOS function 13 (Disk Reset):

```
mov ah,0dh
int 21H
```

- Node 2 opens the shared file for read-only, performs a seek relative to the end of file, and reads the records just added by node 1.

The Disk Reset function call should never be performed while files are open for write, or data may be lost. The file-sharing method above was alluded to in the EtherShare manual, but the details were not mapped out. We used it in a networked real-time data-acquisition-and-display application involving six IBM PCs and found it reliable.

Ray Duncan
Marina del Rey, CA

Yes, DOS function 13 does have the effect described. In the environment described by Duncan it is obviously an effective means of managing a file that changes size. However, in most cases, the use of pre-allocated files and the 3Com semaphore system provides a more versatile means of ensuring data integrity since it allows simultaneous read/write access to all PCs on the network.

Since Duncan indicates that his is a real-time system, I would expect the computational and I/O overhead of the semaphore approach to be prohibitive. Usually that overhead will be negligible.

—William Colsher

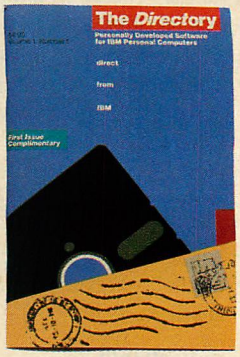
DEAR PCTJ

Thank you for your questions and comments. We appreciate all the letters—complimentary and critical.

Please be sure to include your name and address and send editorial correspondence to us at this address:
PC Tech Journal
Suite 211
The World Trade Center
Baltimore, MD 21202.
Or contact us through the Source ID, STY682

All letters become the property of PC Tech Journal. We reserve the right to edit them in the interest of clarity and our space limitations.

Personally Developed Software



For 36 interesting programs packaged and marketed in a way other publishers should well note, *PC Tech Journal* names IBM's Personally Developed Software for IBM Personal Computers its February 1985 Product of the Month.

The series includes eight entertainment programs, nine in education, 14 productivity programs, and five in business. All are neatly listed in the small booklet pictured here, The *Direct*ory, and available by mail. The programs appear to have been developed by individual authors as part of IBM's long-standing software submission program.

PC Tech Journal is impressed with the pricing of these products. In most cases, we felt the products were offered at about one-half the price they might have commanded through retail distribution channels.

IBM is also offering six Value Packages, combinations of programs from within a family, at a substantial savings.

IBM has chosen a simple but effective packaging scheme for these products, which realizes the value of economy of scale. Each program in the series (except for the Value Packages) comes in an identical folder. A cut-out on the front cover reveals a diskette label that identifies the product, its family, and the minimum system requirements for its operation. The diskette (or diskettes) is held in place on a card inserted in a pocket on the inside of the rear cover. Start-up instructions are printed on the front of the pocket.

A special program called G is responsible for guiding the user toward successful operation of the program and also manages viewing and printing of the documentation. G permits the documentation to be conveniently viewed on the screen. The table of contents is displayed first; the user may move the cursor to highlight a desired section and then read that part directly. Once in the document, the user may scroll forward or backward, or return to the table of contents.

Although the technical staff at *PC Tech Journal* has not tested every program in the series, those we have tested have worked smoothly. Other sources report positive experiences. The findings to date indicate that the whole series is of above average quality. This is astonishing for such a large set of programs, all released at the same time.

IBM has offered quality programs at reasonable prices through a company direct marketing program. The catalog is well done and inviting. The products are well-packaged and interesting. All things considered, Personally Developed Software is a model for other publishers to emulate.

PRODUCT NAME
The *Direct*ory
Personally Developed Software for IBM PCs

COMPANY
International Business Machines Corporation

ADDRESS
P.O. Box 3280, Wallingford, CT 06494

TELEPHONE
1-800-IBM-PCSW; 1-203-237-4504 (Alaska & Hawaii)

PRICES
The Entertainment Family
Backgammon ($19.95)
Blackjack ($19.95)
Crypto-Mania ($19.95)
Kaleidoscope ($14.95)
Roadrunner Rescue ($19.95)
3-K Trivia ($19.95)
Word Seeking ($19.95)
Zuran ($19.95)
The Education Family
Adventures With Decimals, Fractions, and Negative Numbers ($44.95)
Adventures With Decimals ($24.95)
Adventures with Fractions ($24.95)

PRICES (continued)
Adventures With Negative Numbers ($24.95)
Algebra Tutor ($29.95)
Beyond basic BASIC ($19.95)
Matrix Madness ($19.95)
Morse Code Drills ($14.95)
These United States ($19.95)
The Productivity Family
DOS File Tracker ($19.95)
File Facility ($19.95)
PC Palette ($39.95)
Personal Computer Picture Graphics ($29.95)
Phone Directory On-Line ($24.95)
Print Buffer On-Line ($19.95)
Phone Directory & Print Buffer On-Line ($34.95)
SciCalc ($24.95)
Select-A-Font ($19.95)
The StarProof Bridge ($14.95)
STATLIB 1 ($149.95)
STATLIB 2 ($149.95)
Utilities I ($19.95)
Utilities II ($19.95)
The Business Family
Executive Phone Directory ($34.95)
With message feature ($39.95)
Build your own ($149.95)
Portfolio Management System ($99.95)
Project Planning and Scheduling ($149.95)

Available for IBM PC

What C did for Programming
Mark Williams has done for C Programming

The C Programming System from Mark Williams

MWC86 gets your C programs running faster and uses less memory space than any other compiler on the market. Then *csd*, Mark Williams' revolutionary C Source Debugger, helps you debug faster. That's The C Programming System from Mark Williams Company.

MWC86

MWC86 is the most highly optimized C compiler available anywhere for the DOS and 8086 environment. The benchmarks prove it! They show MWC86 is unmatched in speed and code density.

MWC86 supports large and small models of compilation, the 8087 math coprocessor and DOS 2.0 pathnames. The compiler features common code elimination, peephole optimization and register variables. It includes the most complete libraries. Unlike its competition, MWC86 supports the full C language including recent extensions such as the Berkeley structure rules, voids, enumerated data types, UNIX* I/O calls and structure assignments.

Quality is why Intel, DEC and Wang chose to distribute MWC86. These industry leaders looked and compared and found Mark Williams to be best.

User Friendly

MWC86 is the easiest to use of all compilers. One command runs all phases from pre-processor to assembler and linker. MWC86 eliminates the need to search for error messages in the back of a manual. All error messages appear on the screen in English.

A recent review of MWC86 in *PC World*, June, 1984, summed it up:

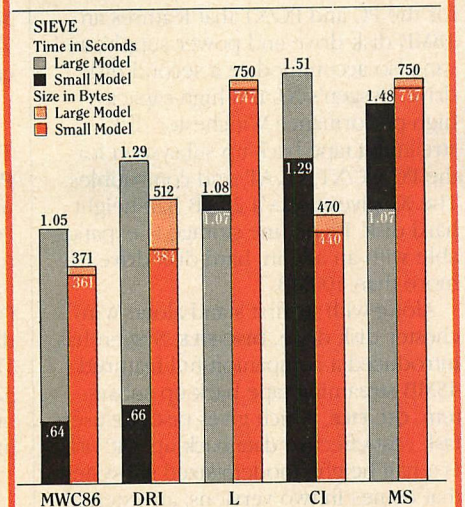
"Of all the compilers reviewed, MWC86 would be my first choice for product development. It compiles quickly, produces superior error messages, and generates quick, compact object code. The library is small and fast and closely follows the industry standard for C libraries."

csd C Source Debugger

Mark Williams was not content to write the best C compiler on the market. To advance the state of the art in software development, Mark Williams wrote *csd*.

csd C Source Debugger serves as a microscope on the program. Any C expression can be entered and evaluated. With *csd* a programmer can set tracepoints on variables and expressions with full history capability and can single step a program to find bugs. The debugger does not affect either code size or execution time. *csd* features online help instructions; the ability to walk through the stack; the debugging of graphics programs without disturbing the program under test; and evaluation, source, program and history windows.

csd eases the most difficult part of development — debugging. Because *csd* debugs in C, not assembler, a programmer no longer has to rely on old-fashioned assembler tools, but can work as if using a C interpreter — in real time.



The C Programming System from Mark Williams now supports the following libraries:

Library	Company
Windows for C	Creative Solutions
Halo	Media Cybernetics
PHACT	PHACT Associates
The Greenleaf Functions	Greenleaf Software
Btrieve	SoftCraft

The C Programming System from Mark Williams

The C Programming System from Mark Williams delivers not only the best C compiler for the 8086 but also the only C source level debugger. That's why it does for C programming what C did for programming. The Mark Williams C Programming System gives the programmer the MWC86 C compiler and the *csd* C Source Debugger for only $495. Order today by calling 1-800-MWC-1700. Major credit cards accepted.

Technical support for The Mark Williams C Programming System is provided free of charge by the team that developed it.

Mark Williams Company
1430 W. Wrightwood Ave.
Chicago, IL 60614

CIRCLE NO. 186 ON READER SERVICE CARD

*Unix is a Trademark of Bell Laboratories.

Hardware, software, and other developments for the PC

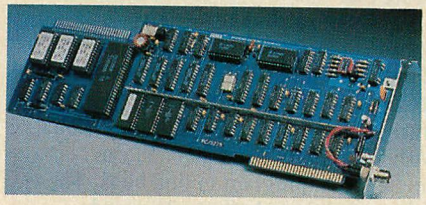
The Sysgen product family

HARDWARE

A **117-megabyte, internal hard-disk drive** for use with the PC/AT has been announced by **PC's Limited**. The drive drops directly into the AT box and works off the existing controller and power supply. It has an average access time of less than 30 milliseconds and a dedicated head landing and shipping zone. The hard-disk drive unit has Whitney-type, ferrite read-write heads and a rotary voice coil activator with a dedicated, closed-loop, track-following servo system. Price not available.
PC's Limited, 7801 North Lamar, #E-200, Austin, TX 78752; 512/452-0323
CIRCLE 497 ON READER SERVICE CARD

AST Research, Inc. has introduced the **COLOSSUS**, a high-capacity, high-performance 67MB Winchester disk drive with a 30-millisecond access time and a 60MB, one-fourth-inch, streaming-tape cartridge back-up unit housed in a single stand-alone enclosure. Standard COLOSSUS features and capabilities include either an 8-bit-PC or 16-bit-AT host adapter, support for the SCSI high-performance, peripheral bus interface, use of the de facto industry-standard ST-506/ST-412 disk drive interface with a five-megabits-per-second transfer rate, support for both streaming and file-by-file back-up operations at 5MB per minute, and use of the QIC-36 and QIC-24 tape interface and format standards, making it possible for tape cartridges from COLOSSUS to be used on other one-fourth-inch tape systems supporting standard interfaces. $7,449.
AST Research, Inc., 2121 Alton Avenue, Irvine, CA 92714; 714/863-1333
CIRCLE 496 ON READER SERVICE CARD

IRMA, the micro-to-mainframe interface from **Digital Communications Associates, Inc.,** has been adapted to bring its emulation and file transfer capabilities to the PC/AT. IRMA now allows an AT to emulate an IBM 3278/79 terminal and to transfer data to and from a mainframe. An IRMA-equipped AT would have this mainframe access while retaining its stand-alone computing powers, allowing the user to communicate with the mainframe using software resident on the AT. Price: $1,195.
Digital Communications Associates, Inc., 303 Technology Park, Norcross, GA 30092; 404/448-1400
CIRCLE 493 ON READER SERVICE CARD



IRMA for the PC/AT

Sysgen, Incorporated has announced the addition of four new products to its line. **DISK I/O** is a six-slot expansion box for the PC and PC/XT that features an 18MB disk drive and power supply, and can also accommodate a second disk drive. Sysgen's **XL** is a high-capacity, high-performance Winchester disk/streaming tape back-up subsystem for the PC, PC/XT, PC/AT, and compatibles. The XL gives users a 20MB half-height hard disk. Its on-line storage is expandable with an add-in, hard-disk drive to more than 100MB.

Along with its first stand-alone Winchester disk drive, **DISC-FILE**, Sysgen has introduced a companion full-featured 45MB streaming tape back-up subsystem, **QIC-FILE**, which gives DISC-FILE users fast, cost-effective data back-up. QIC-FILE is a half-height, modem-sized subsystem that comes in two versions: an external model and an internal model. Prices: DISC I/O, $1,995; XL, $3,295; DISC-FILE, $2,995; QIC-FILE, external model, $1,495, internal model, $1,395.
Sysgen, Incorporated, 47853 Warm Springs Blvd., Fremont, CA 94539; 415/490-6770
CIRCLE 492 ON READER SERVICE CARD

Eastman Kodak Company has introduced a new family of audiovisual production tools designed for use in conjunction with computer peripherals and other business machines. The new **Instagraphic** products include a CRT slide imager, copy stand, slide mounter, and color slide film. The slide imager can be used to produce color slides of graphics displays directly from the PC's CRT. Color transparencies can be produced from a Kodak Instagraphic camera back fitted to a video image recorder (VIR) that is wired directly to a PC. A VIR performs the same function as a CRT, but produces a slightly sharper picture with somewhat truer color. Prices: CRT slide imager, $339.00; copy stand, $549.00; slide mounter, $74.99; color slide film, $12.95 for 10 exposures.
Eastman Kodak Company, 343 State Street, Rochester, NY 14650; 716/724-3169
CIRCLE 494 ON READER SERVICE CARD

The new **3270 PC** and **3278/79 PLUS PC** connections have been designed for use with the IBM PC/AT operating under PC-DOS 3.0 and 3.1, according to its manufacturer, **CXI**. This enables AT users to have up to five host sessions, a PC-DOS session, and two notepads all displayed simultaneously on the screen. The 3270 PC connections provide full 3270 PC emulation and allow use of one or more attached parallel or serial printers as host-addressable IBM 3278

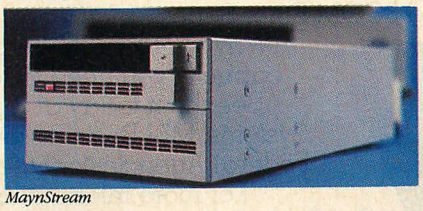
Color 400

printers. The 3278/79 PC connectors provide 3278/79 emulation as well as control unit terminal mode emulation of the 3270 PC. Both of the connectors come with window management software and an expanded file transfer capability. They are available either in coaxial or modem form. Prices: 3270 PC, $1,595; 3278/79 PLUS PC, $1,145.
CXI, 3606 W. Bayshore Road, Palo Alto, CA 94303-4229; 415/424-0700
CIRCLE 475 ON READER SERVICE CARD

Megaram-PC, a high-speed, solid-state, nonrotating memory system for the PC, PC/XT, and PC/AT, has been introduced by **Imperial Technology**. The system interfaces to the computer as if it were a disk, but with access times that are 267,000 times faster than a floppy disk and 75,000 times faster than a fixed disk, and with a transfer rate of 0.95MB per second. Housed in a 3.8-inch high desktop enclosure, the Megaram-PC offers truly random access to a range of capacities from 0.5MB to 8MB that is expandable in 0.5MB increments. It is delivered with an I/O driver that, once installed in the operating system, makes the Megaram-PC compatible with all applications software. Prices: .05MB, $1,700; additional increments of .05MB, $680 each.
Imperial Technology, Inc., 831 S. Douglas Street, Suite 102, El Segundo, CA 90245; 213/536-0018
CIRCLE 484 ON READER SERVICE CARD

A high-resolution, desk-top image scanner for office automation, designated the **MS-200**, has been introduced by **Microtek Lab, Inc.** General applications of the MS-200 would be graphics and pictures input to computer systems in business, industry, government, academic, and other large or small organizations. It accepts documents up to 8½ inches by 24 inches, digitizes the image at 200-pixels-per-inch resolution, and transfers the image to host computer memory. Switch-selectable scanning modes includes text, picture, and mixed modes. For quantities over 500, MS-200 sells for $1,000 each (OEM price).
Microtek Lab, Inc., 17221 South Western Avenue, Gardena, CA 90247; 213/538-5369
CIRCLE 483 ON READER SERVICE CARD

Maynard Electronics has introduced the **MaynStream**, a full-streaming, self-contained tape drive that backs up the PC, PC/XT, PC/AT, Portable PC and MS-DOS-compatible hard drives. MaynStream can back up a 10MB drive in approximately three minutes and offers users the option of backing up the entire disk or changed files only. The cartridge handles 26.7MB of data at a transfer rate of 90KB per second. $1,895.
Maynard Electronics, 430 East Semoran Blvd., Casselberry, FL 32707; 305/331-6402
CIRCLE 489 ON READER SERVICE CARD


MaynStream

Hayes Microcomputer Produts has announced the **Smartmodem 2400** that transfers data across regular telephone lines at speeds of up to 2400 bps and offers a link between mainframe and personal computers. The Smartmodem 2400 meets the CCITT (Consultative Committee on International Telephone and Telegraph) V.22 international standard; once it has been approved by telecommunications authorities in major markets, the modem can provide a link for data communications around the world. The new modem operates both synchronously and asynchronously, which enables it to link terminals, personal computers, microcomputers, mainframes, and information utilities.

Hayes has also released **Smartcom II 2.0**, an upgrade of its telecommunications software that includes XMODEM protocol, VT52 and CT102/100 terminal emulation and batch commands for automatic data transmission at specified times. Smartcom II 2.0 protects passwords, tests the modem without going on-line, and lets users switch easily from voice to data transmission during one phone call. Prices: Smartmodem 2400, $899; Smartcom II 2.0, $149.
Hayes Microcomputer Products, Inc., 5923 Peachtree Industrial Blvd., Norcross, GA 30092; 404/449-8791
CIRCLE 482 ON READER SERVICE CARD

A high-resolution color/graphics board, designed for the PC, has been introduced by **Sigma Designs, Inc**. The **Color 400** provides 100-percent software compatibility with standard IBM graphics software and offers 640-dots-by-400-lines, all point-addressable graphics in noninterlaced mode. It has a programmable scan rate from 25.5 KHz to 31 KHz, making it compatible with the newer high-resolution monitors. Color 400 offers 16 colors from a palette of 256, and is compatible with 40-by-25 and 80-by-50 modes. Bundled with the Color 400 is a custom version of PC Paintbrush; an optical mouse by Mouse Systems is also included. Prices: Color 400 with PC Paintbrush and Mouse, $895; Color 400 alone, $795.
Sigma Designs, Inc., 2023 O'Toole Avenue, San Jose, CA 95131; 408/943-9480
CIRCLE 495 ON READER SERVICE CARD

HIKE-UP YOUR AT!

WHY WAIT? GET 100% COMPATIBLE HARD DISK AND TAPE BACKUP FOR YOUR IBM® PC AT™ FROM MOUNTAIN TODAY.

Hike-up your basic AT to an enhanced AT with Mountain's internal hard disk. Installs in minutes using the AT's disk controller, power supply, and DOS 3.0. Choose from 20MB, 35MB or 120MB (formatted).

And don't forget our AT internal tape that lets you back up your irreplaceable files with complete confidence. It's fast. Economical. Easy-to-use. And installs in minutes. Complete with menu-driven software. Choose from 27MB or 60MB (formatted).

Both come network-ready to support Novell, Orchid PCnet,® NESTAR and 3COM Ether Series.

Why wait for IBM's hard disk when you can get just the capacity you need today. Plus reliable tape backup.

See them today at Entre (1-800-HI-ENTRE), Morris Decision Systems, MBI Business Centers, Office Technology Plus (Federal Agencies only), and other leading computer stores.

For more information, write: Mountain Computer, Inc., 300 El Pueblo Road, Scotts Valley, CA 95066. Or call (408) 438-6650. TWX 910-598-4504.

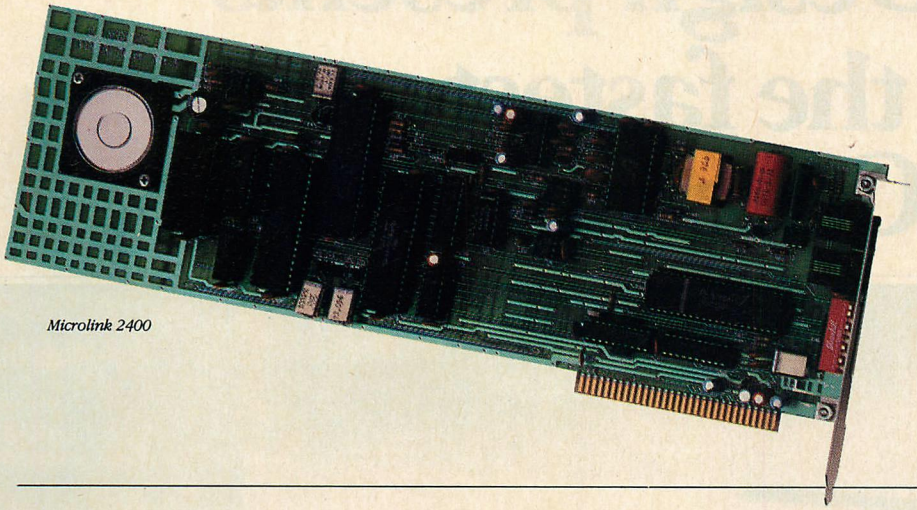
Mountain®

THE PEAK OF PERFORMANCE

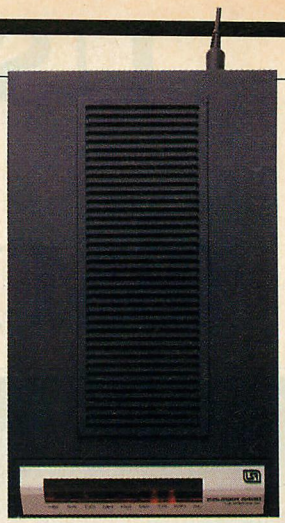
1-800-458-0300 (in California, 1-800-821-6066)

In Canada, call Parity Plus Inc. (416) 673-3321.
In France, call Compusol 01-530 07 37.
In Sweden, call Nordic Data 08-730 50 50.

CIRCLE NO. 154 ON READER SERVICE CARD



Microlink 2400



Courier 2400

Kontron Electronics has announced the availability of a software package that enables Intel development tools to execute on the PC. The Kontron **PC Universal Development Interface** (UDI) works with Kontron's recently announced **Personal Instrumentation** (PI) product. The PC UDI software package and PI interface card, when installed in a PC, create a complete Intel development environment for designing, debugging, and testing of 8086/88 applications programs. The Kontron PI interface card provides the link required to connect the PC to Kontron's line of real-time, in-circuit slave emulator subsystems.

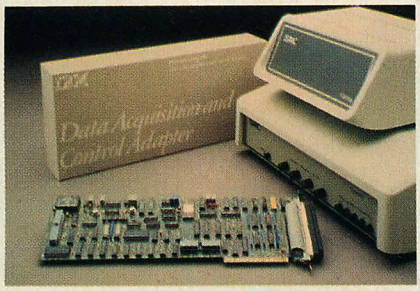
Also introduced was an emulator port that turns the PC/XT into a universal development system. The **Kontron PC Interface** (KPCI) package consists of hardware and a set of development support software tools, including cross-assembler, linker, emulator software, and additional CP/M utilities. Prices: UDI, $500; PI, $1,500, KPI, $1,500.
Kontron Electronics, 630 Price Avenue, Redwood City, CA 94063; 415/361-1012

CIRCLE 491 ON READER SERVICE CARD

Cyborg Corporation announces the availability of new **IBM Data Acquisition and Control Adapters** for the PC, PC/XT, PC/AT, and PC Portable. It is directly hardware- and software-compatible with Cyborg's **ISAAC** family of computer/instrument interface systems. Designed to automate analytical instruments and industrial processes, the adapter is ideally suited for temperature monitoring and control, vibration measurements, and physiological testing. Applications range from basic and applied research to materials testing and full-scale process monitoring.

Also announced was the availability of a new **ISAAP application note** describing the use of LOTUS 1-2-3 and Symphony to analyze scientific and engineering data with the Adapter, the ISAAC 41-I, 91-I, or ISAAC 2000. Prices: IBM Data Acquisition and Control Adapter, $1,275; ISAAP application note, included.
Cyborg Corporation, 55 Chapel Street, Newton, MA 02158; 617/964-9020

CIRCLE 487 ON READER SERVICE CARD



ISAAC

FASTORE recently introduced **SoliDISK**, an advanced, solid-state, nonvolatile mass storage system designed for use with the PC and compatibles. Offering a new alternative to floppy and hard disk users, it provides the user with quick and dependable access to any program. The diskless, high-performance system incorporates software security protection as well as a tolerance of a wide variety of environments. SoliDISK uses 64KB and 128KB EPROMS with permanent storage of 192KB to 384KB. Its onboard EPROM programmer uses the latest intelligent programming algorithms and, with associated software, can copy a double-sided diskette onto the SoliDISK in less than 45 minutes. Once programmed, it can be used as a high-speed disk with an access time of two milliseconds. Prices: single-density board, $650; double-density board, $1,250; quad-density unit, $2,500.
FASTORE, 200 Danbury Road, Wilton, CT 06897; 203/834-1975

CIRCLE 481 ON READER SERVICE CARD

The stand-alone **Courier 2400** and integral **Microlink 2400** are new 2,400-baud, auto-dial modems introduced by **U.S. Robotics, Inc.** The two products feature full AT-command set compatibility and several unique functions and commands for enhanced ease of use and efficiency. They provide complete dial-up communications capabilities at rates of 2,400, 1,200, and 300 bits per second. Microlink 2400, designed for integral use with the PC and compatibles, includes the company's Telpac communications software. The Courier 2400 may be used with any computer or ASCII terminal equipped with an RS-232 serial interface. Prices: Courier 2400, $895; Microlink 2400, $895.
U.S. Robotics, Inc., 1123 West Washington Blvd., Chicago, IL 60607; 312/733-0497

CIRCLE 490 ON READER SERVICE CARD

Data Electronics, Inc. has introduced a new 20MB tape cartridge subsystem for the PC/XT and compatibles. The **SL-6¼-Inch Tape Cartridge Subsystem** features the File Selectable tape drive that enables users to retrieve and work with files in the same way they do with floppy or hard disks. In addition to primary use of the tape back-up system for save and restore, the file-selectable application offers an advantage in that operations involving data files can continue in the event of hard-disk failure or other types of data loss. Price: $1,995.
Data Electronics, Inc., 10150 Sorrento Valley Road, San Diego, CA 92121; 619/452-7840

CIRCLE 480 ON READER SERVICE CARD

U.S. Design presents the fastest IBM® PC Storage System.

The only thing it doesn't do is make you wait for delivery.

Now you can get a high-performance alternative to the mass storage systems you're considering for your IBM PC, PC XT, PC AT, and compatible systems. We can ship it to you today. It's called the U.S. Design PC-870. It's incredibly fast (average accesses as low as 20 ms.) thanks to the world's most advanced cache controller technology. And it's fully compatible.

Compatibility: we deliver.

Just for starters, it boots. Using only one slot for both disk and tape, you can handle all your storage needs from 60 megabytes up to 240 megabytes. You can even define two logical MS-DOS disks. Try that with another drive!

When it comes to backing up the disk, our 20 megabyte cartridge tape drive is easier to use than an IBM floppy. Our software looks like the IBM backup utility, but in addition it automatically creates the proper directory structures, and allows selective backup by filename or date.

Reliability: delivering the highest quality.

Because we made the PC-870 fast by making it smart, it delivers high performance that you can trust. Our exclusive adaptive cache controller technology makes the PC-870 a perfect solution for network applications. The design has been tested and proven with thousands of units already placed in corporate environments where dependability is absolutely essential.

For immediate delivery.

We can ship to qualified customers overnight. So think fast. We'll act fast. Call us now toll-free, and ask John Dunn, VP-Sales and Marketing, for all the details. 1-800-368-2811.

U.S. DESIGN

5100 Philadelphia Way, Lanham, Maryland 20706
(301) 577-2880 • (800) 368-2811 • TWX 710-826-0417

CIRCLE NO. 210 ON READER SERVICE CARD

IBM is a registered trademark of International Business Machines Corporation.

Coax Exchange

The **DT2803,** a low-cost, video frame grabber and display board for the PC and the PC/XT, has been announced by **Data Translation, Inc.** The DT2803 captures a 256-by-256-by-6, black-and-white image, and the RGB output handles 64 colors by 64 intensities and includes cursor control. In combination with the also-announced **VIDEOLAB** software package, the DT2803 is ideal for applications in industrial robotics, inspection, and assembly, medical imaging, and graphic arts. Via standard RS-170 (60Hz) or CCIRR (50Hz) compatible input signals, the DT2803 allows the user to digitize a video field every 1/30 second. VIDEOLAB, written in C, takes advantage of the fast execution times and convenience of high-level languages. Prices: DT2803, $1,495; VIDEOLAB, $995.
Data Translation, 100 Locke Drive, Marlboro, MA 01752; 617/481-3700
CIRCLE 479 ON READER SERVICE CARD

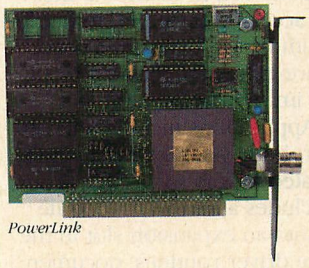
DT2803

From **Forte** comes the **ForteGraph for IRMA**, which provides mainframe graphics capabilities to IBM PCs, using the IRMA 3278/79 terminal emulator. ForteGraph allows the PC to act as a fully functional IBM color graphics terminal linked to an IBM mainframe. The IRMA board is required. Price: $,1595.
Forte Data Systems, 2205 Fortune Drive, San Jose, CA 95131; 408/945-9111
CIRCLE 454 ON READER SERVICE CARD

Intelligent Technologies has introduced an integrated hardware/software package providing IBM communications via coaxial cable. **Coax Exchange** enables a PC to emulate an IBM 3278/89 terminal in a variety of mainframe environments, including IMS, VM/CMS, MVS/TSO, CICS, and IDMS. Consisting of a printer circuit board and software on diskette, the package utilizes coaxial cable to link a PC with an IBM 3274 or 3276 cluster controller. Price: $1,195.
Intelligent Technologies International Corporation, 151 University Avenue, Palo Alto, CA 94301; 415/328-2411
CIRCLE 488 ON READER SERVICE CARD

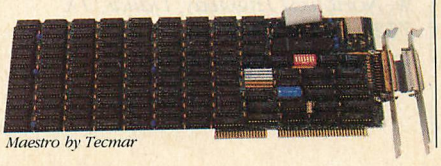
Daisy Systems Corporation announces the **Personal Logician AT**, a computer-aided engineering system for the IBM PC/AT. The workstation includes fully integrated design-entry capacilities, as well as the Personal Daisy Logic Simulator for design verification. The Personal Logician AT, which is based on the PC/AT, includes a high-performance, high-resolution graphics controller, a 1,022-by-840 pixel monitor, 640KB RAM, a 20MB hard disk, and a 1.2MB floppy disk drive. Prices: complete system, $25,000; add-on package for existing PC/AT computers, $20,000; color version for an additional $10,000; optional EtherNet capability, $2,500 per workstation.
Daisy Systems Corporation, 700 Middlefield Road, Mountain View, CA 94043; 415/960-0123.
CIRCLE 456 ON READER SERVICE CARD

Tecmar has announced the release of two new products. The **PowerLink** is a micro-to-mainframe communications board that is designed to emulate IBM 3278/79 terminals, allowing a user's PC to access 3270 applications on a mainframe while concurrently running PC applications. With its software, it is intended to fill systems integration needs of the Fortune 500 and middle-market firms, as well as major health and education institutions. The design of the interface board features direct memory access for instantaneous screen updating and enables quick transfer of data between the mainframe and PC.



PowerLink

The **Maestro** multifunction board is designed specifically to enhance the PC/AT. The memory expansion on Maestro provides two memory blocks. The first block upgrades the AT to a maximum of 640KB of RAM, which is the maximum amount of memory usable by the DOS 3.0. The second block can be upgraded to 2MB of memory using 64KB chips or 256KB chips. This will be important in a multiuser environment. PowerLink, $1,099; Maestro, from $529 for 0KB plus 2 MB to $3,989 for 128KB plus 2MB.
Tecmar, Inc., 6225 Cochran Road, Solon, OH 44139-3377; 216/349-0600
CIRCLE 478 ON READER SERVICE CARD



Maestro by Tecmar

PCVISION Frame Grabber

Imaging Technology Inc. designs, manufactures, and markets a modular family of plug-compatible, real-time image-processing modules and subsystems called the **IP-512** for OEMs, systems integrators, and end users. One of the company's products, **PCVISION Frame Grabber**, is a board-level video digitizer and display module that is plug-compatible with the PC, PC/XT, and PC/AT. The module captures an RS-170 video signal (TV standard) at a rate of 30 frames per second, stores the image in an on-board 512-by-512-by-8-bit frame memory, and simultaneously displays the stored image on a monitor.

The image stored in the frame memory can be accessed by the PC to implement algorithms for image enhancement, feature extraction, pattern recognition, and image transmission and archiving. Applications include teleconferencing, robotic vision, x-ray analysis, and automated factory inspection. The package includes a hardware module that plugs into an expansion slot in the PC, software driver routines, documentation, and cables. Prices: $3,495 for color; $2,995 for monochrome.
Imaging Technology Inc., 600 W. Cummings Park, Woburn, MA 01801; 617/938-8444
CIRCLE 485 ON READER SERVICE CARD

Applied Creative Technology, Inc. has introduced the **Systemizer**, a print buffer with 16KB or 64KB of memory; it allows up to 15 computers to share printers. An optional third connection allows several Systemizers to link together. $299 and up.
Applied Creative Technology, Inc., 2156 W. Northwest Highway, Dallas, TX 75220; 214/556-2916
CIRCLE 455 ON READER SERVICE CARD

SOFTWARE

Action Diagrammer is a new programming design tool based on a structured, graphic representation of the actions comprising a software program as a hierarchy of bracketed blocks. At the highest overview level a program is shown as a single bracket. Control flow and processing logic are grouped and bracketed in deeper levels of the hierarchy. A color monitor adds clarity to diagrams. **Database Design, Inc.** has designed Action Diagrammer to supply control structure syntax automatically in English, COBOL, PL/1, FORTRAN, C, Pascal, and many fourth generation languages. Price: $495; a demonstration diskette is available for $25.
Database Design, Inc., 2020 Hogback Road, Ann Arbor, MI 48104; 313/971-5363
CIRCLE 477 ON READER SERVICE CARD

A runtime version of REVELATION has been announced by its creators, **Cosmos, Inc. Runtime REVELATION** is the same as REVELATION without R/DESIGN, REVELATION's menu-driven applications generator, and R/BASIC, the software's structured procedural language. $150.
Cosmos, Inc., 19530 Pacific Highway South, Suite 102, Seattle, WA 98188; 206/824-9942
CIRCLE 468 ON READER SERVICE CARD

Control Data Corp. has added two enhancements to its microcomputer-based engineering design system: software for simulation analysis of logic and timing, or **SALT**; and the **Alphamap** cell library, designed by **Alphatron**. SALT allows an engineer to use the PC/XT or a compatible to simulate the timing and logic of the circuit from the simple transistor to the more complex functional level. Users may enter data serially with the system's Network Description Language. The software automatically computes propagation delays. Other features include automatic computation of fan-in and fan-out delays, multiple checkpoint restart, recompilation of networks through incremental compilation, automatic classification of transistors as uni- or bidirectional, automatic determination of critical path, output assertions, and multi-level simulation. $3,500.

Alphamap provides the engineer with VLSI design capability and allows him to lay out working large-scale integrated circuits. The Alphamap Design and Development System, which includes functional cell decals and software to check the layout, is $3,495. A design handbook and decals for manual design are available for $995. Alphatron will produce tooling, prototypes, and production test programs for about $35,000.
Control Data Corporation, 2500 Mission College Blvd., Santa Clara, CA 95054; 800/538-5332, or in California, 800/672-3522
CIRCLE 471 ON READER SERVICE CARD

Alphatron, 10351 Bubb Road, Cupertino, CA 95014; 408/446-1491
CIRCLE 470 ON READER SERVICE CARD

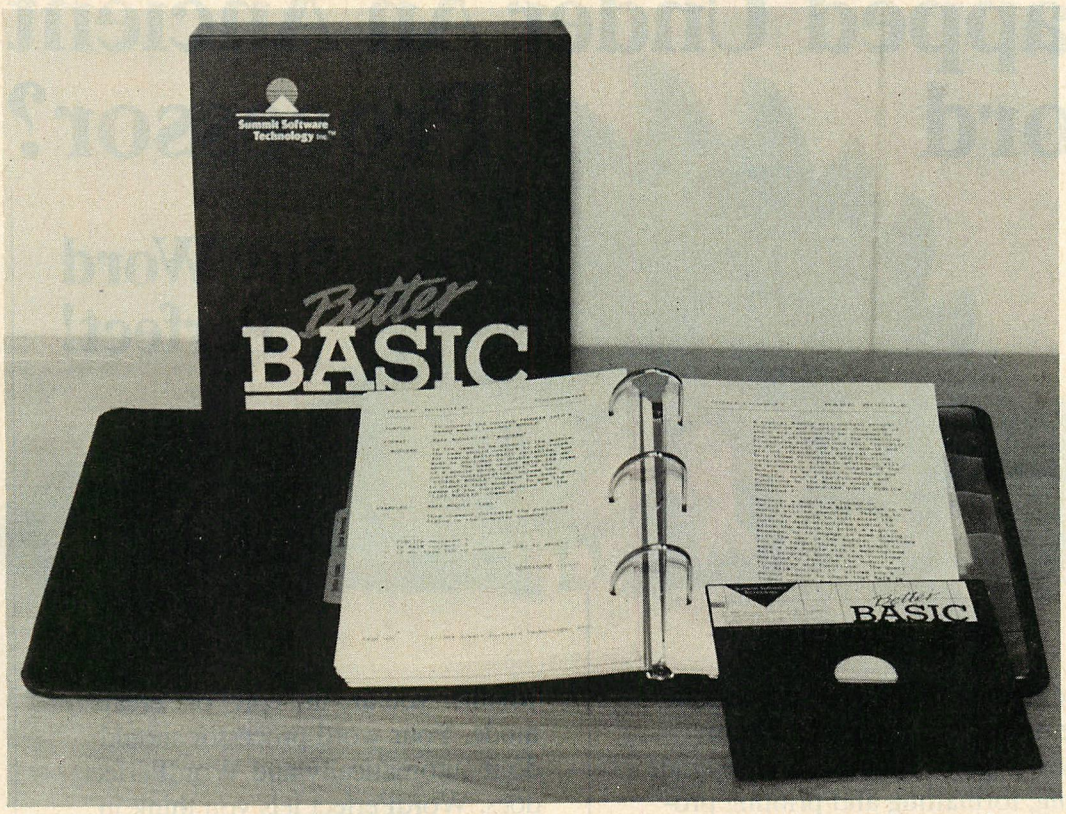
Rational System, Inc. has released **Instant-C**, a C language interpreter for the PC and other systems running MS-DOS or CP/M-86. Instant-C is said to execute programs 20 to 50 times faster than interpreted BASIC does, and several times faster than FORTH or P-Code-based compilers do. Its built-in-screen editor displays source code errors with the cursor set to the trouble spot. It has more than 200 diagnostic messages. Its debugger works in C language. $500.
Rational Systems, Inc., P.O. Box 480, Natick, MA 01760; 617/653-6194
CIRCLE 467 ON READER SERVICE CARD

WHY WOULD ANY SANE PERSON SPEND $199 FOR A BetterBASIC SYSTEM WHEN DOS's IS FREE?

HERE ARE 10 REASONS:
TEST YOUR SANITY

1. Full support for 640K memory **2.** Structured language with BASIC syntax **3.** Separately compiled program modules **4.** Speed: FAST **5.** Extensibility (Make your own BASIC.) **6.** User-defined procedures and functions **7.** Built-in windows support **8.** Interactive programming language based on an incremental compiler **9.** 8087 math support **10.** Runs on IBM PC, IBM PC/XT and compatibles

SANE PROGRAMMERS ORDER BetterBASIC NOW



CALL (617) 235-0729. Summit Software Technology, Inc.™
P.O. Box 99, Babson Park, Wellesley, MA 02157
Price: $199 Sample Disk $10
8087 Math Module: $99 Runtime System: $250

NOW AVAILABLE FOR THE TANDY 2000 & 1200

Better BASIC™

BetterBASIC is a trademark of Summit Software Technology, Inc. IBM PC, IBM PC/XT and PC/DOS are trademarks of International Business Machines Corp. MS-DOS is a trademark of Microsoft Corp.

MasterCard, VISA, P.O. Checks, Money Orders, and C.O.D. accepted.

CIRCLE NO. 177 ON READER SERVICE CARD

How Can You Avoid Getting Trapped Under An Ancient Word Processor?

Buy Word Perfect!

Once upon a time, word processors were monstrous things. Dot commands, page orientation, and separate editing, formatting and printing programs turned them into lumbering beasts. Only a well-educated programmer would dare don his armor and tackle such a beast — not a pleasant task for a modern secretary, executive, or writer.

Then came WordPerfect and the beast was slain.

WordPerfect was designed to work for you not against you. WordPerfect has no command language to complicate your writing. Pressing a single key is all it takes to

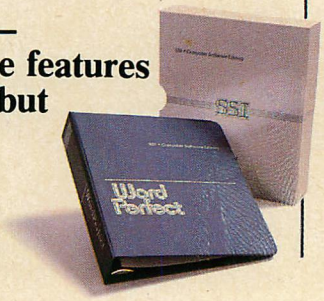
bold, underline or center.

When writing, you don't want to worry about page formatting, making room for headers or footers, or whether you are in "edit" or "create" mode. Your word processor should do it automatically and WordPerfect does. WordPerfect lets you think in terms of ideas, not pages. It is simple enough that you quickly forget about the mechanics and your writing flows easily.

So if you don't want to be caught under a word processing monster, try WordPerfect. We're certain it will improve the quality of your writing.

WordPerfect

You'll love it — not only for the features we've built in, but also for the antiquities we've left out.



SATELLITE SOFTWARE INTERNATIONAL

288 WEST CENTER STREET, OREM, UTAH 84057 (801) 224-4000 TELEX 820-618

SSI

CIRCLE NO. 236 ON READER SERVICE CARD

Cornerstone

Sound Presentations from **Communication Dynamics** is a new graphics software package, built on the IBM Graphics Development Toolkit, that can be used with IBM's recently introduced Professional Graphics Display. It will produce color slides, overhead transparencies, and hardcopy. The software comes with template libraries containing maps of continents and major U.S. metropolitan markets, charting formats, and a collection of business-oriented symbols. Price: $250.
Communication Dynamics, 8532 SW Saint Helens Drive, Wilsonville, OR 97070; 503/682-0651
CIRCLE 469 ON READER SERVICE CARD

Infocom has launched its business products division with a relational database system for the nonprogrammer. Called **Cornerstone**, the product is designed specifically for business managers without technical expertise and therefore does not require learning any complex commands or programming skills. To build a database with Cornerstone, the user answers a series of questions. Cornerstone includes an interactive report writer, and it also lets users add notes and comments to their files whenever they want. These notes, which can be a few words or several pages, can be searched for specific words. The product has a built-in help system that is context-dependent. Cornerstone runs on the PC, PC/XT, PC/AT, and compatibles. It runs under Topview, and can also be used in conjunction with many widely used spreadsheets and word processors. A convert capability allows users of other database products to transfer existing files into the system. Cornerstone comes with a tutorial, owner's handbook, and a sample database application. $495.
Infocom, Inc., 55 Wheeler Street, Cambridge, MA 02138; 617/492-1031
CIRCLE 476 ON READER SERVICE CARD

Disk P.M. (Disk Preventive Maintenance), a new product that performs preventive maintenance on hard and floppy disks for the PC and compatibles, has been introduced by **Digital Pathways, Inc.** Disk P.M. diagnoses problems, automatically condenses hard or floppy disks, restores order, rebuilds damaged directories, recovers damaged files, locks out faulty areas, and copies system information to disks that refuse to boot. It can run unattended while condensing. Disk P.M. can read data files that have been damaged. It moves the files to another location on the disk, puts them into a readable format, and reconnects them, allowing the data file to be saved. Price: $49.95.
Digital Pathways, Inc., 1060 East Meadow Circle, Palo Alto, CA 94303; 415/493-5544
CIRCLE 486 ON READER SERVICE CARD

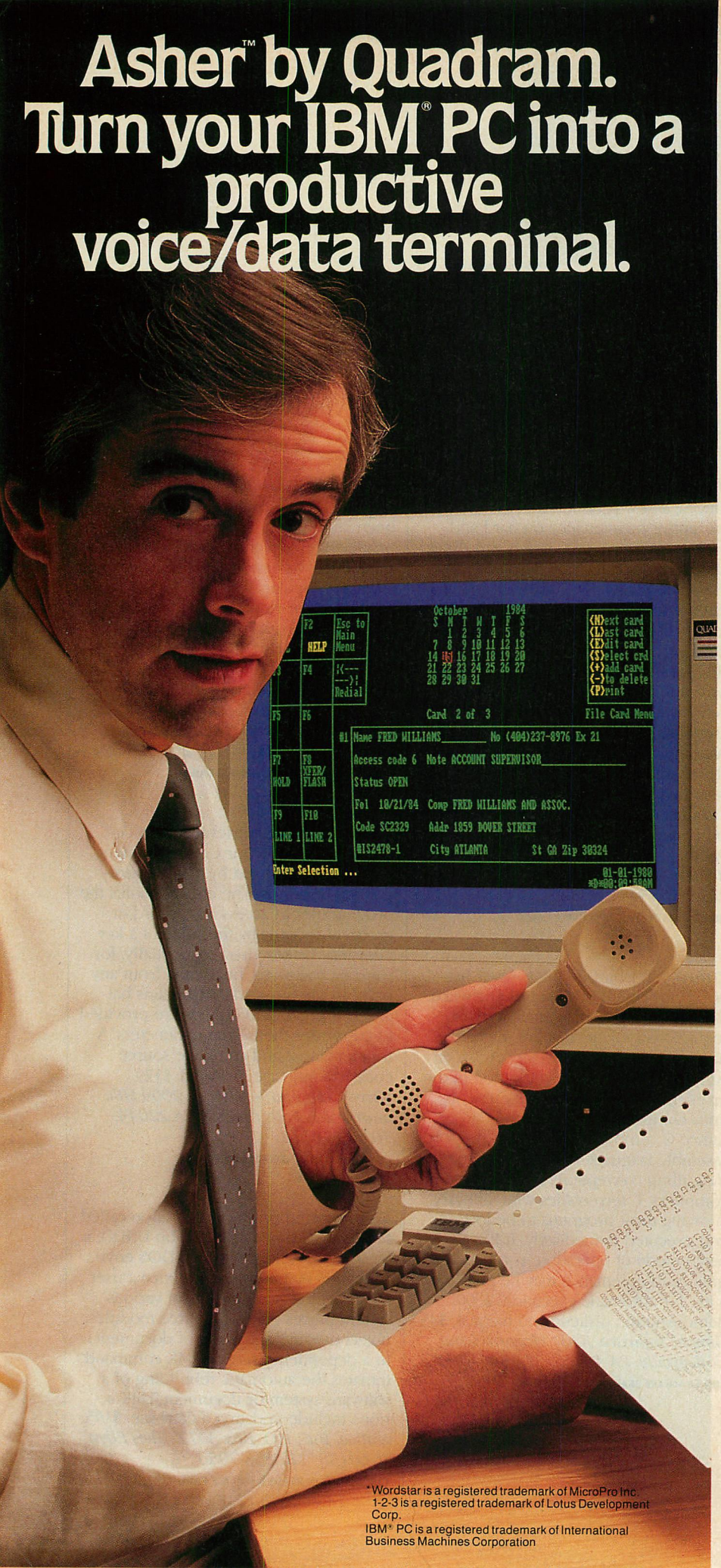
C-terp, a C language interpreter, has been introduced by **Gimpel Software**. C-terp converts C programs in to a sequence of tokens that are interpreted, providing feedback of errors. A screen editor shows the precise location of the problem. C-terp supports library routines compiled with either the C86 or Lattice C compilers. It will also support assembly language code conforming to the calling conventions of these two compilers. C-terp comes with a library, allowing most programs to be loaded and executed immediately. It has a batch mode of operation in which a sequence of commands may be placed in a file for automatic execution. $300; demonstration diskette and manual, $45.
Gimpel Software, 3207 Hogarth Lane, Collegeville, PA 19426; 215/584-4261
CIRCLE 466 ON READER SERVICE CARD

A computer-aided design software package called **Design Board 3D Link** has been announced by **MEGA CADD**. The product converts three-dimensional designs into two-dimensional drawings by allowing popular two-dimensional drafting software packages for personal computers to read the drawing files from MEGA CADD's Design Board 3D software. Designers can then use the same database created during the initial design phase in later work phases. $295.
MEGA CADD, Inc., 401 Second Avenue South, Seattle, WA 98104; 206/623-6245
CIRCLE 460 ON READER SERVICE CARD

CBTREE, a record retrieval system designed specifically for C programmers, has been announced by **Micro Computing Services**. The utility enables the programmer to access any record or group of records by the value of a key. Files may be searched sequentially, forward or backward, beginning from any point in the file. CBTREE maintains balanced B-Trees, and utilities are provided to customize the B-Tree parameters. A user's manual and complete source code example are included. $179.
Micro Computing Services, 2009 Hileman Road, Falls Church, VA 22043; 703/893-0118
CIRCLE 465 ON READER SERVICE CARD

An automated product generation called **LMK** (also known as MAKE on UNIX) has been developed for MS-DOS environments by **Lattice, Inc.** Once a user has specified relationships of various system pieces (such as source modules, object modules, chapters of document) in a dependency file, an LMK command initiates the automatic rebuilding of a software system or document. LMK actions include updating a database. $195.
Lattice, Inc., P.O. Box 3072, Glen Ellyn, IL 60138; 312/858-7950
CIRCLE 463 ON READER SERVICE CARD

Asher™ by Quadram. Turn your IBM® PC into a productive voice/data terminal.



Asher. Communications management for your computer and you.

Organize your personal and business communications quickly, effectively. With Asher by Quadram. Snap Asher inside your PC for a system that works the way you do. Asher begins by partitioning system memory. Load all your programs (Wordstar,* Lotus 1-2-3,* you name it) just once then toggle between them when your work calls for it. With Asher, program selection is just a keystroke away.

Asher turns your PC into a full-featured telephone.

Asher lets your PC become a computerized telephone, too. Packed with all the features you'd find in an advanced electronic key phone. Place voice calls through the Asher handset (works just like a desk phone) and exchange files or send a screen to other systems through the Asher modem. With Asher, you can even talk and send data during the same call.

Plus, Asher comes with an electronic calendar to track your daily schedule (It even beeps you when an appointment time arrives). And a card file that lets you record telephone numbers and data from all your business transactions.

Effective communications management for any business environment.

Asher's power and ease-of-use (most functions are executed with a single keystroke) make it ideal for any business environment. Because Asher integrates your system. Now you can place a voice call, call up a data base on your screen, and transmit information to another system all in the same operation. It's effective communications management at its best.

™

QUADRAM QUALITY

Visit your local authorized Quadram dealer today and see how Asher (and that famous Quadram Quality) can go to work for you. Asher. Powerful communications management for your personal computer and you.

QUADRAM™
An Intelligent Systems Company

4355 International Blvd./Norcross, Ga. 30093
(404) 923-6666/TWX 810-766-4915 (QUADRAM NCRS)

INTERNATIONAL OFFICES

Interquadram Ltd. 442 Bath Road, Slough, England SL16BB
Tel: 6286-63865 Tlx: 847155 Auriema G **Interquadram GmbH**
Fasanenweg 7, 6092 Kelsterbach, West Germany Tel: 6107-3089
Tlx: 417770 Seva G **Interquadram s.a.r.l.** 41, Rue Ybry, 92522 Neuilly
Tel: 758-1240 Tlx: 630842 Iso Bur **Chevco Computing** 6581 Kitimat
Road, Mississauga, Ontario, Canada L5N-2X5 Tel: 416-821-7600

CIRCLE NO. 211 ON READER SERVICE CARD

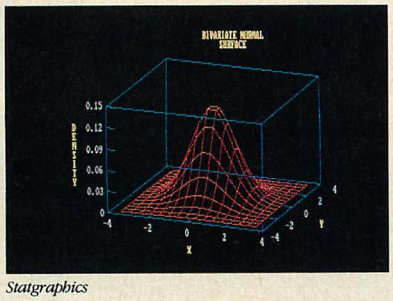
*Wordstar is a registered trademark of MicroPro Inc.
1-2-3 is a registered trademark of Lotus Development Corp.
IBM® PC is a registered trademark of International Business Machines Corporation

Screen Sculptor

The Software Bottling Company of New York has developed a product that allows the user to design customized input screen programs in IBM BASIC, IBM Pascal, and Turbo Pascal. Called **Screen Sculptor**, the software provides a word processor-type editor with a color selection screen, special character selection screen, a repeat-last-character facility, and color painting. After a screen has been "drawn," Screen Sculptor will generate source code based on the screen design. Price: $125.
The Software Bottling Company of New York, 29-14 23rd Avenue, Long Island City, NY 11105; 718/728-2200
CIRCLE 498 ON READER SERVICE CARD

Microtec Research has developed an extended, high-performance Pascal compiler for the Intel 80286 family of microprocessors running MS-DOS. **Professional Pascal** produces optimized code. Optimizations include common subexpression elimination, retention and re-use of register contents, dead-code elimination, short/long jump, constant folding, range analysis and redundant range-check elimination, and short-circuit evaluation of Boolean expressions. The compiler supports five memory models and ROMable code for embedded applications. Its language extensions are designed so that programs can be changed from ANSI-standard to Professional Pascal. More than 200 error and warning diagnostics are supplied, with the source code line and column number of the error reported. Runtime library routines and utilities are included: string and heap operations, a portable I/O library, access to the command line and environment variable, and system interfaces. A set of UNIX-like utilities come with the package. A single computer use license is $895.
Microtec Research, Inc., P.O. Box 60337, Sunnyvale, CA 94088; 408/733-2919
CIRCLE 461 ON READER SERVICE CARD

A joint publishing venture by **STSC, Inc.** and **Statistical Graphics Corporation** has resulted in the release of **STATGRAPHICS**, a software package integrating comprehensive statistical analysis with high-resolution color graphics. STATGRAPHICS offers more than 250 functions for interactive data analysis and data management, including analysis of variance, regression analysis, experimental design, quality control procedures, and forecasting and time series analysis. The product supports pen plotters and high-resolution color graphics. It requires a minimum of 320KB, a graphics adapter card, and two floppy disk drives. Price: $695.
STSC, 2115 E. Jefferson Street, Rockville, MD 20852; 301/984-5000
CIRCLE 473 ON READER SERVICE CARD



Statgraphics

A compiler for dBASE III has been introduced by **Nantucket, Inc.** With **Clipper**, applications will run 2 to 20 times faster than they would under the standard, interpretive version of dBASE III, according to its developers. Clipper generates an executable relocatable element that cannot be decoded to the original source language, ensuring protection of code. Debugging is assisted with compiler-generated error messages. Additionally, existing bugs in the interpretive version of dBASE III have been removed. $695.
Nantucket, Inc., 20456 Pacific Coast Highway, Malibu, CA 90265; 213/456-7315
CIRCLE 459 ON READER SERVICE CARD

QUICKCODE III, which is claimed to almost triple the power of dBASE III, has been announced by **Fox & Geller**. QUICKCODE III is the successor to the QUICKCODE program generator. It expands the power of dBASE III by using a jumbo file feature that lets users create data files eight times the size of normal dBASE III files. It includes on-screen field calculations, a report library for sorted reports, and on-line help. $295.
Fox & Geller, Inc., 604 Market Street, Elmwood Park, NJ 07407; 201/794-8883 or 800/221-0156
CIRCLE 462 ON READER SERVICE CARD

Two library supplements for the Microsoft FORTRAN Compiler have been announced from **Alpha Computer Service**. **STRINGS and THINGS**, which is written in assembly language, contains support for character manipulation, the SHELL command, batch processing, music creation, and general register manipulation. The source code for several demonstration programs is included. STRINGS and THINGS requires a minimum system configuration of a PC or PC/AT with 256KB of memory and PC-DOS 2.0 or later. Price: $69.95.
Alpha Computer Service, P.O. Box 2517, Cypress, CA 90630; 714/894-6808
CIRCLE 453 ON READER SERVICE CARD

50 to 500MB ... 5X Speed Removable Hard Disk ... Minicomputer Performance for Your IBM PC

Our newest Maverick rigid disk subsystems for your IBM PC, -XT, -AT or compatible can give you 50 to over 500MB capacities and speeds three to five times faster than the current Winchesters. Everything you need for network file servers or large data base management applications.

The latest additions to our Maverick line are fully compatible with your existing hardware and software and support a large selection of drives with both fixed and removable media. Designed for both VARs and end-users, they are furnished complete with our Maverick SMD disk controller and software for PC DOS™, QNX™ and other popular operating systems — everything you need to plug-in and run.

And you'll run trouble-free with our Maverick subsystems. With Interphase's one-year warranty and the types of disk drives normally found on minicomputers and mainframes, high reliability is assured. Menu-driven installation, booting from hard disk, and compatibility with most LANs are other key features. We also offer a Maverick configuration that makes it easy for you to perform your own system integration.

So don't just sit there wishing your PC could do more! Maverick subsystems are available now! Call your nearest Interphase dealer. Or contact us at 2925 Merrell Road, Dallas, Texas 75229, telephone 214/350-9000, Telex 73-2561 (TELESERV) DAL.

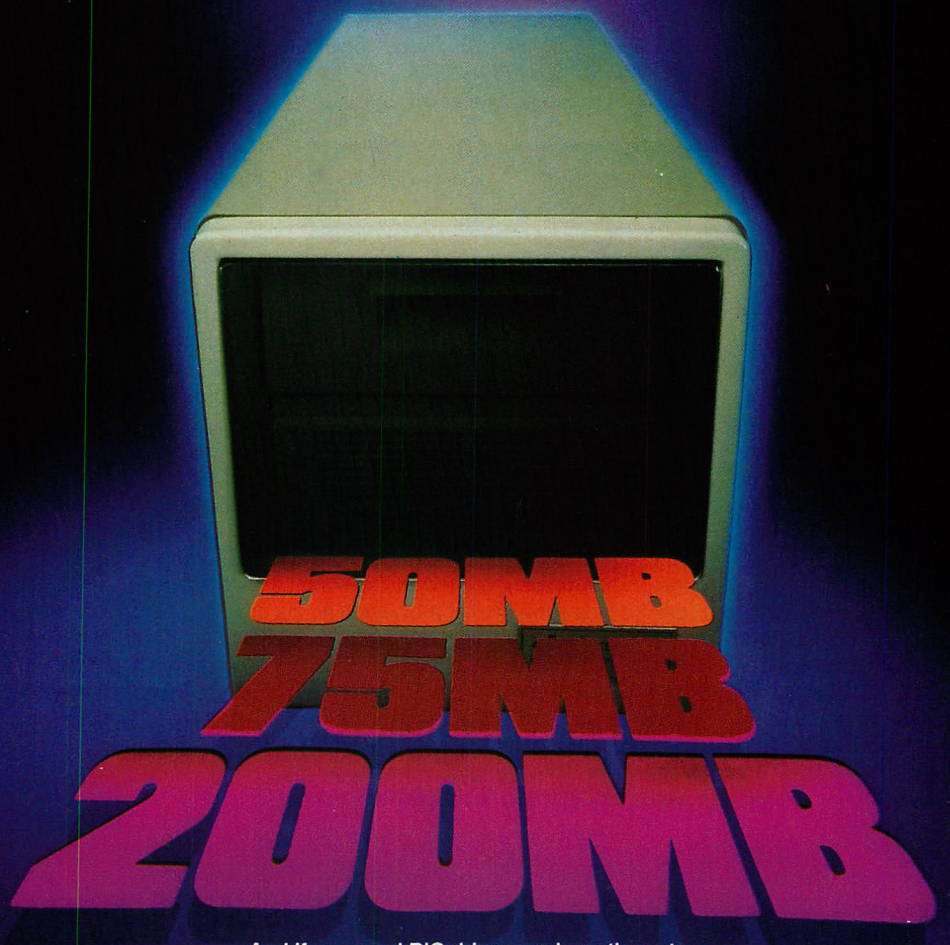
Dealer Inquiries Invited.

In The Southeast Call:
Southeastern Data Products
Perrowville Road Route 2
Forrest, Va 24551
(804) 525-2494

INTERPHASE corporation

Trademark of IBM • Trademark of Quantum Software Systems Ltd.

In The Northeast Call:
Chase Technologies, Inc.
375 Sylvan Avenue
Englewood Cliffs, NJ 07632
(201) 894-5544



And if you need BIG drives, we have those too.

CIRCLE NO. 142 ON READER SERVICE CARD

Presenter PC

SoftCall (AT&T)

DICOMED has announced a software package with charting and graphics capabilities to create slides and overheads of up to 8,000 lines of resolution. **Presenter PC** has the ability to tap DIF files to generate an array of charts and graphs. It provides 64 colors that can be used simultaneously on an image, or the user may select one of eight preselected color schemes for automatic coloring of charts and graphs. Two fonts are provided, and lines, circles, rectangles, and triangles can be manipulated in real-time. The software is menu-driven and a single-line help feature is displayed at all times with a full page of documentation available for the currently active function. Price: $995.

DICOMED, 12000 Portland Avenue South, Minneapolis, MN 55400-0246; 612/885-3000

CIRCLE 472 ON READER SERVICE CARD

AT&T has introduced **SoftCall**, a menu-driven software program to work with the company's new Model 4000 modem. The SoftCall communications package does not require the user to remember any codes or special instructions, but explains itself as it is used. The new software provides a call history log that keeps a record of the time, date, duration, telephone number, and baud rate of all data cells made and received. It can store and automatically dial up to 30 telephone numbers. A remote operations feature allows the user to use the auto-answer mode so callers can leave messages in a special file or have access to the computer's files by entering a password. Price: $79.95.

AT&T Consumer Products, 5 Wood Hollow Road, Parsippany, NJ 07054; 201/581-5185

CIRCLE 464 ON READER SERVICE CARD

A Program Development Environment for BASIC, called **PDEB**, has been introduced by **Jackson Software** for users of the IBM BASIC interpreter and compiler. PDEB provides a full-screen editor especially designed for BASIC, which maintains line numbers automatically. The interpreter and compiler can both be invoked with a single keystroke. PDEB commands are designed to be similar to commands of the BASIC interpreter. Price: $39.95

Jackson Software, 1729 Mayflower Drive, Carrollton, TX 75007

CIRCLE 458 ON READER SERVICE CARD

GrafTalk 3.27 recently made its debut with several new features added to the business graphics package, manufactured by the **Redding Group**. Among the features is Disk Image, which increases printer output speed by as much as 300 percent. It also enables the user to take advantage of a printer's full resolution and color capabilities, regardless of any limitations of the screen resolution. GrafTalk 3.27 produces bubble charts in which three variables can be graphed at once. It also features a dual labeling capability for the two Y axes on line and symbol plots. $450.

Redding Group, Inc., 609 Main Street, Ridgefield, CT 06877; 203/431-4661

CIRCLE 474 ON READER SERVICE CARD

Interface Technologies Corporation has announced a **Modula-2** development system that includes a fast compiler, multiple windows, on-line help, color graphics and sound support modules, and functions for direct access to PC-DOS facilities. Price: $249.

Interface Technologies Corporation, 3336 Richmond, Houston, TX 77098; 713/523-8422

CIRCLE 452 ON READER SERVICE CARD

RAYTRONICS has introduced three software products for programmers. **C-CHAIN** is a C language support package for expanding memory usage of the PC, allowing multiple independent programs to be loaded and chained. Each program can occupy 64KB of code and 64KB of data; program overlays are supported. Data areas can be common or independent. A utility is provided for calling any loaded program and passing up to 10 parameters and return status.

PC-DEBUG is a unique program that allows programs to be debugged independent of the operating system using a serial terminal connected to the main system. Multiple EXE files can be loaded and run; code can then be examined—and modified. Up to 10 breakpoints are supported; it includes complete disassembly and trace facilities.

Also announced is **ANDROMEDA**, a real-time, multitasking system designed for automation and monitoring applications. Although PC-DOS compatible, **ANDROMEDA** can run independent of DOS. Split-screen and windowing are supported for both console and serially-connected terminals; customizable modules are provided for terminals and printers. An optional, integrated **PC-DEBUG** module can be used to debug and monitor the programs and data of multiple real-time tasks. Prices: **C-CHAIN**, $95; **PC-DEBUG**, $95; **ANDROMEDA**, $295.

RAYTRONICS, 7392 Trade Street, San Diego, CA 92121, 619/566-7515

CIRCLE 451 ON READER SERVICE CARD

INTRODUCING
Interface Technologies' Modula-2 Software Development System

The computer press is hailing Modula-2 as "the next standard in programming languages." Modula-2 combines the strengths of Pascal with the features that made C so popular, like independent compilation and direct hardware control.

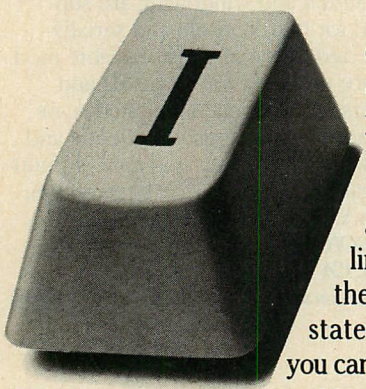
But until today, no company offered a Modula-2 system that made the development of software fast, easy and efficient. Now, though, there's a new tool at your disposal.

The fast, powerful tool for programmers

The breakthrough is here: Interface Technologies' new Modula-2 Software Development System for the IBM® PC, XT, AT and compatible computers to give programmers the same quantum leap in productivity spreadsheets and word processors gave to end-users. It can reduce monotonous wait time, will dramatically increase speed, help stop thoughtless mistakes, and free you to become more creative in virtually all of your programming efforts.

How to speed input and eliminate 30% of errors

Thirty percent of programming mistakes are syntax errors and simple typos in the program structure. Our "syntax-directed" Modula-2 editor does away with these time-consuming headaches once and for all.



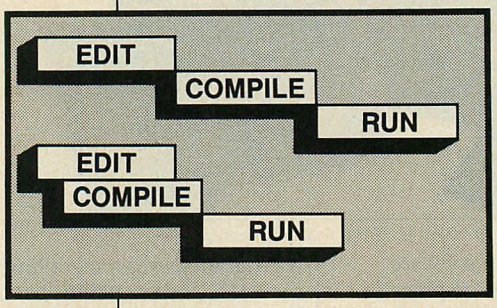
Enter complete statements with one keystroke.

It speeds input by cutting manual typing as much as 90%, letting you enter statements with a single keystroke. For example, if you type a capital "I" to begin a line, the editor completes the logical "IF THEN" statement automatically, so you can concentrate on what you want to program, rather than concentrate on what you're typing.

The editor locks out errors, finishing statements and procedures in perfect accord with the standardized rules of Modula-2. It also indents and formats your text automatically, making programs easy to read and maintain, an important feature on big projects.

And if you leave an undefined variable or data type, the editor detects the mistake and gives you the option of on-line "help" to correct it. No other programming text editor offers you so much innovation at any price.

How to turn "wait time" into "work time"



It not only has a faster compiler, it also saves time by compiling <u>while</u> you edit.

The vast majority of programming time is spent waiting, and the biggest slowdown is most often with compilers.

THE ANATOMY OF A

Our compiler turns wait time to work time with a new innovation that lets you compile in the "background."

With background compilation, your program is automatically compiled into object code line by line as you work, every minute you spend writing or editing a Modula-2 program!

When you're finished editing, all that's left for the compiler is a quick mopping up job that generates optimized native code in a single pass.

How quick is "quick"?

Thanks to background compilation and the fact that the compiler itself is so fast, Interface Technologies' compiler turns 100 lines of typical Modula-2 text into optimized machine code in *under five seconds.*

Plus the Interface compiler produces compact code with execution speed superior to that produced by any other Modula-2 compiler on the market.

How to do two things at once

Along with the background compiler and syntax-directed editor, which can save you hours every day and make you more productive, Interface Technologies' Software Development System gives your monitor

windows so you can refer to one file while you edit another simultaneously, saving you even more time.

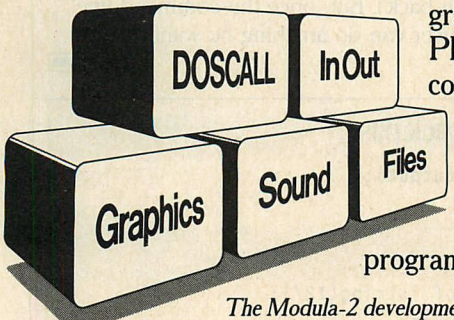
Concurrent editing of two or more files is especially useful when doing programming work that's intended for separate compilation, and Interface Technologies has the only Modula-2 system on the market that provides you with this helpful benefit for developing software.

Work with multiple files faster, easier in windows.

How preprogrammed modules speed development

One of the advantages of Modula-2 is that it lets you build large, reliable programs quickly, by linking together many smaller "building-block" modules.

The development system's toolkit of precompiled program modules includes the standard Modula-2 library, and adds exclusive link-and-run modules for direct calls to the operating system, sound, and color graphics support. Plus you get low-cost updates from the Interface Technologies fast-growing library of new programming modules.



The Modula-2 development system's toolkit of ready-made modules turns big programs into smaller projects.

Increase productivity for $249

Interface Technologies' Software Development System is fast, powerful and unlimited. It works so well that it's the same tool Interface Technologies is using to write business and consumer applications in Modula-2.

For $249, you get the syntax-directed editor and compiler, linker, module library and tutorial that will have even modestly experienced programmers writing in Modula-2 in days. And you have full rights to your work; there's no license fee for programs you develop with the Interface Technologies system.

You can use it on any IBM® PC, XT, AT or compatible computer with two double-sided, double-density floppy drives and 320K RAM diskette.

You get a thoroughly indexed, comprehensive user's manual and free telephone support from Interface Technologies. But the most important thing you get is the future, and *the programming language of the future is Modula-2*, and now it's easier than ever.

For more information, or to order the Modula-2 Software Development System, call 1-800-922-9049 today. In Texas, call (713) 523-8422.

You can also order or request further information by mail. Just fill out the coupon below and send it in. Act today and receive your system soon.

Operates on the new IBM PC AT, as well as the PC, XT, and all other IBM-compatible computers.

BREAKTHROUGH

NAME _____

ADDRESS _____

CITY _____ STATE _____ ZIP _____

PHONE _____

PLEASE CHECK ONE:

☐ AMERICAN EXPRESS ☐ VISA ☐ MASTERCARD

☐ CHECK ENCLOSED

CHARGE ACCOUNT NUMBER

☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐

_____ _____

EXPIRATION DATE SIGNATURE

PLEASE SEND ME _____ COPIES @ $249 EACH.

INTERFACE TECHNOLOGIES CORPORATION

3336 RICHMOND, SUITE 200, HOUSTON, TX 77098

Texas residents, add 6.125% Sales Tax. **PC/2**

INTERFACE TECHNOLOGIES

MODULA-2 SOFTWARE DEVELOPMENT SYSTEM

IBM is a registered trademark of International Business Machines Corporation.

CIRCLE NO. 123 ON READER SERVICE CARD

TECH NOTEBOOK 30

Grabbing Parameters

Retrieving command line arguments from IBM Pascal 2.0

ROBERT B. STAM

The simplest way to retrieve arguments from the command line is to use Pascal's program parameters (see the *IBM Pascal Compiler 2.0 Reference Manual*, Volume 1). This works well with a fixed number of parameters in a simple format.

Another approach is needed, however, when more flexible argument structures are desired. Pascal's program parameters are strictly positional, and none of them are optional (if one is omitted, the user will be prompted for it). Another kind of parameter a programmer might want to use is the keyword parameter, which is optional and can appear in any position. To handle other kinds of arguments, the program must analyze the command line itself. The question is how to retrieve the command line intact.

One approach is to declare a program parameter of type LSTRING (variable length string). The Pascal runtime system will copy the command line to this variable. The only problem is that the runtime system deletes the first character from the command line. The most likely reason for this is that the first character is usually a blank. COMMAND.COM strips away the name of the program before copying the rest of the command line. Unless some other character terminates the program name (for example, a /), the first character after the program name will be a blank.

If the programmer is using switches, he will need to see the / to know that the next character is a switch. To get the entire command line, including the first character, some more subtle techniques will have to be called into play. The key to solving this problem is to locate the original command line, and then read it directly from there.

The command line is stored in the Program Segment Prefix (PSP) built by COMMAND.COM, described in the *DOS 2.0 or 2.1 Reference Manuals*. This is referred to as the unformatted parameter. It is located at offset 80H in the PSP, and is 128 bytes long. The first byte is set to the number of characters in the command line (maximum 127), followed by the command line itself.

This just happens to be the internal representation for LSTRINGs in Pascal. What the user needs to do is superimpose the definition of a Pascal LSTRING on top of these memory locations. The problem is to declare a variable that will be superimposed on top of the command line. Because the data already exists, a declaration must be used that is not going to allocate new memory.

This is the purpose of reference types. Pointer variables do not create new data, they point to existing data. The user needs a pointer variable pointing to the command line. Because the command line is outside of the Pascal runtime data space, standard reference types cannot be used.

IBM Pascal provides a systems programming extension to reference types called addresses (ADR and ADS). The address type needed here is the ADS segmented machine address; ADS pointers can access any location in memory.

The ADS pointer can be made to point to the original command line by setting the segment to point to the PSP, and the offset to point to offset 80H. This is not hard because ADS variables can also be used as records with two word fields. The .S field is the segment; the .R is the relative offset.

But where is the PSP? It is not well-documented, but IBM Pascal has a global word variable called CESXQQ that has the segment address of the PSP. The value of this global variable can be copied to the .S field of the ADS pointer.

The ADS pointer can be used to process the command line *in situ*, but it is best to copy it over to a local variable for further processing. One reason is that working with pointers can get confusing. A more important reason is that the compiler can generate more efficient code for local variables.

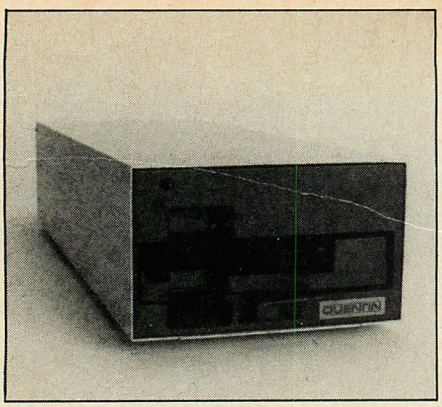
The program in the listing shows how to use these techniques. This program does nothing with the command line (other than echo it back). But, once the command line has been located, the user can do anything he wants with it.

LISTING: echoback.pas

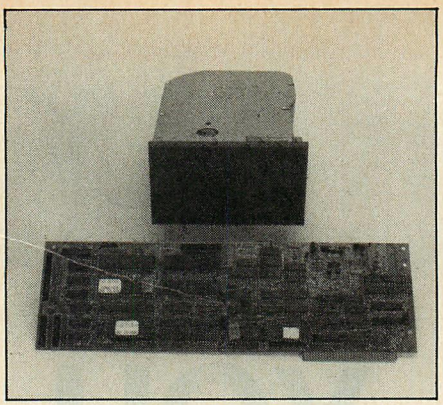
```
program echoback (output);

var

commandline : ads of lstring(127);
CESXQQ [extern] : word;
args : lstring(127);

begin

  commandline.S := CESXQQ;
  commandline.R := #80;
  args := commandline^;

  writeln (args);

end.
```

Robert B. Stam is a graduate of Harvard College and is vice-president of Futureware, Inc. of Jackson, Mississippi. He has taught several computer science courses at the University of Costa Rica.

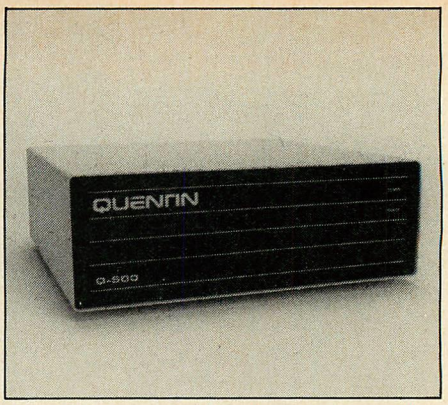
YOU DON'T HAVE TO CHOOSE BETWEEN MEGABYTES AND MEGABUCKS *ANYMORE!*



25 Megabyte Cartridge Streaming Tape System for the IBM-PC and IBM-XT **$995**



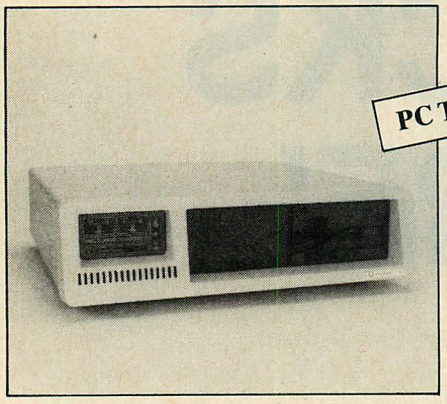
Add-in Winchester Disk System for the IBM-PC. 10 to 40 Megabyte Capacity Starting from **$795**



10 to 32 Megabyte Winchester System for the IBM-PC. The Q-500 Supports Additional Add-on Winchesters to 32 Megabytes and Add-on 25 Megabyte Streaming Tape. Prices from **$1,295**

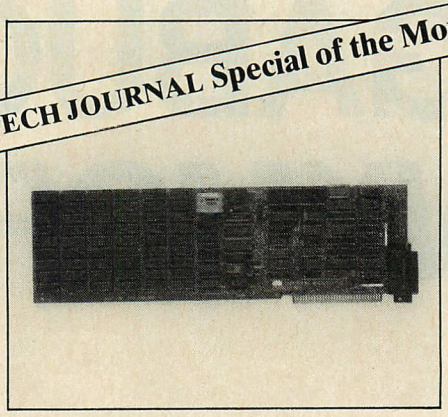
QUENTIN.

BETTER FOR LESS.

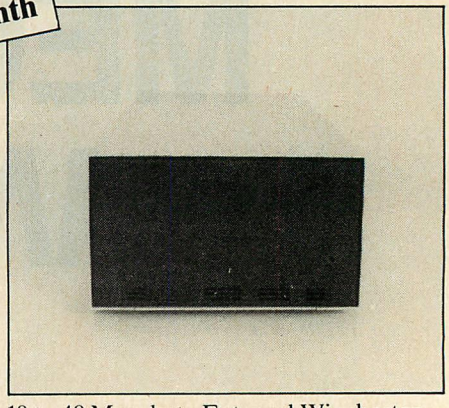


8 Port Expansion Chassis and Winchester Disk System for the IBM-PC. Winchester Capacities from 10 to 150 Megabytes with Optional 25 Megabyte Cartridge Streaming Tape or 5 or 10 Megabyte Removable Winchesters. Prices starting from **$1,695**

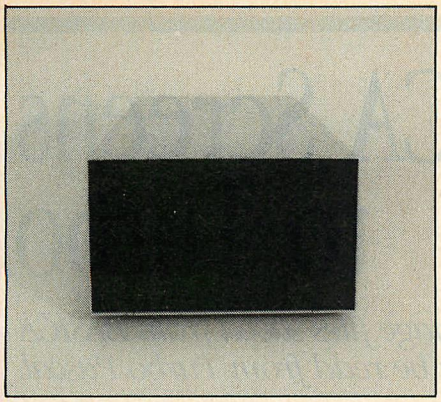
PC TECH JOURNAL Special of the Month



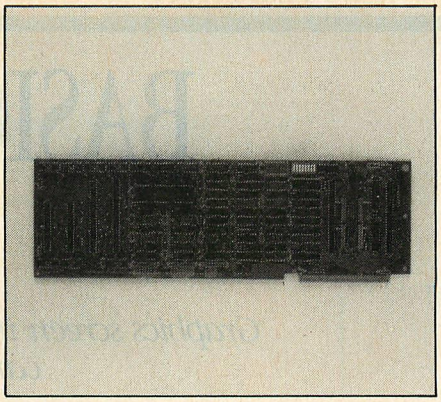
Multifunction Board with Serial and Parallel Port, Real Time Clock/Calendar, and up to 384K Parity Checked Memory (Equivalent to the AST Six Pak) Multicard + with 64K **$195**



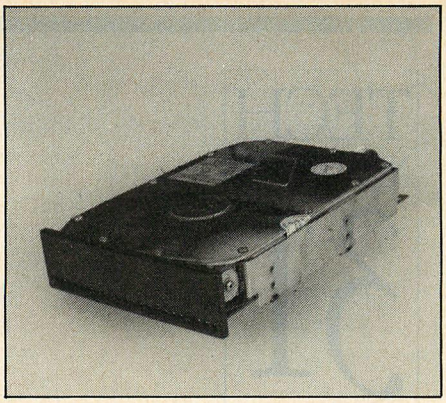
10 to 40 Megabyte External Winchester System for the IBM-PC. Prices starting from **$1,095**



10 to 32 Megabyte External Winchester System for the IBM-XT. Prices starting from **$1,095**



5¼" and 8" Floppy Disk Controller for the IBM-PC and XT **$125**



32 Megabyte Half Height Add-in Winchester for the IBM-AT **$1,395**

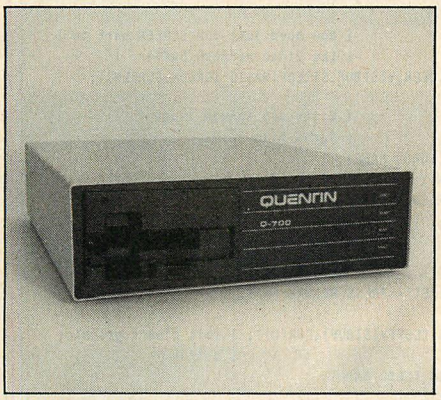
■ ■ DOING THINGS

How do we do it? We cut out the middleman without cutting out the support. Quentins's factory staff of design engineers, manufacturing engineers, and product support technicians are available daily from 8:00 a.m. until 5:00 p.m. to answer your questions and to provide technical assistance for our full line of products.

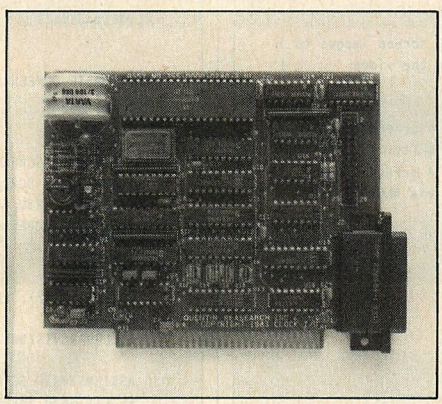
All Quentin products are warrantied for 90 days with optional one and two year extended warranty programs available.
Visa and Mastercard accepted.

1-800-555-1212
*(ask for our **new** toll free number)*

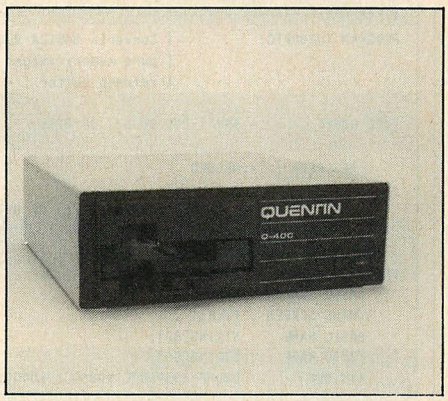
QUENTIN
5308 Canwood
Agoura Hills, Calif. 91301



10 MB to 32 MB Winchester System with Internal 25 MB Streamer Tape System. Available for Both the IBM-PC and the IBM-XT. Prices starting at **$1,895**



Serial Port, Parallel Port, and Real Time Clock/Calendar on Short Slot Card for the IBM-PC, XT, and AT **$95**



25 Megabyte Cartridge Streaming Tape Backup System for the IBM-AT **$1,195**

CIRCLE NO. 189 ON READER SERVICE CARD

TECH NOTEBOOK 31

BASICA Screens to Turbo

Graphics screen image files saved from BASICA can be read from Turbo Pascal.

JEFF DUNTEMANN

Turbo Pascal now provides an alternative to doing graphics programming in BASIC, encouraging some users to stop working in IBM BASICA almost entirely. However, graphics screen images saved to disk from BASICA with BSAVE are not read easily from Turbo Pascal, a problem for using several disks of BASICA screen image files with Turbo. How does BASICA's BSAVE save a binary image to disk?

When BSAVE writes a file to disk, it tacks on a seven-byte header to the front of the file that tells BLOAD the type of file and the size of the its binary image. Turbo Pascal's BLOCKREAD procedure provides the fastest method of getting information from a binary image file to screen buffer RAM, but it *must* begin reading at the start of a disk block. The actual graphics screen image in a BSAVE file begins seven bytes into the disk block. BLOCKREAD loads the header into the graphics refresh buffer ahead of the actual graphics pixel information. The graphics information is offset by seven bytes and looks thoroughly scrambled. So, in order to read a BASICA screen image file properly, the header must be eliminated.

The accompanying listing is a simple translator program that will read a BASICA BSAVE screen image file into a record structure that treats the header and the screen image array as separate data items. The screen image array portion of the record is then MOVEd into the video refresh buffer. The image in the refresh buffer is written out to disk via BLOCKWRITE, minus the seven-byte header. Using the READ procedure to read the BASIC-style file into a record is not as fast as BLOCKREAD, but once the screen image has been read completely into RAM, writing it out again via BLOCKWRITE is lightning fast—faster than BSAVE (due to BASICA's interpreted nature). And reading the screen image back to RAM with BLOCKREAD is faster than BLOAD by a factor of three.

The program displays the screen image file mostly for dramatic impact; actually, the method is not restricted to screen image files. Any file originally saved via BSAVE may be expressed as a record consisting of the seven-byte header and an array of the number of bytes needed to contain the rest of the file. Once the record is read into memory (there should be enough for the entire file) the user can write the array part back to disk minus header.

Jeff Duntemann is a programmer for Xerox Corp. in Rochester, NY.

LISTING: TURBOPIC.PAS

```
{-------------------------}
{      <<<TURBOPIC>>>     }
{    by Jeff Duntemann    }
{    Turbo Pascal V2.0    }
{    Last Update 12/11/84 }
{-------------------------}
PROGRAM TURBOPIC;           { Converts BASICA BLOAD screen images to }
                            { pure memory images of the video        }
                            { refresh buffer                         }

TYPE GBUFF       = ARRAY [0..16383] OF BYTE;    { A graphics screen }
                                                { buffer }
     BASCREEN    = RECORD                       { A screen saved }
                                                { via BLOAD }
                     HEADER : ARRAY[0..6] OF BYTE;
                     SCREEN : GBUFF;
                   END;
VAR BASFILE      : FILE OF BASCREEN;
    BASIC_SCREEN : BASCREEN;
    TURBO_SCREEN : FILE;
    BASIC_NAME   : STRING[80];
    TURBO_NAME   : STRING[80];
    VISIBUF      : GBUFF ABSOLUTE $B800 : $0000;

BEGIN
  WRITE('>>Enter the name of the BASICA-format screen file: ');
  READLN(BASIC_NAME);
  ASSIGN(BASFILE,BASIC_NAME);  { Check if requested file exists }
  RESET(BASFILE);
  IF IORESULT = 255 THEN
    WRITELN(BASIC_NAME,' is not on disk!  Try again...')
ELSE
  BEGIN
    WRITE('>>Enter the name of the Turbo-format screen file: ');
    READLN(TURBO_NAME);
    GRAPHCOLORMODE;                 { Switch to graphics mode }
    READ(BASFILE,BASIC_SCREEN);     { Read the BASICA-saved file }
                                    { to RAM }
    CLOSE(BASFILE);
                                    { Now move just the SCREEN part to }
                                    { the video refresh buffer: }
    MOVE(BASIC_SCREEN.SCREEN,VISIBUF,SIZEOF(BASIC_SCREEN.SCREEN));

                                    { Write out screen image to }
                                    { Turbo file }
    ASSIGN(TURBO_SCREEN,TURBO_NAME);
    REWRITE(TURBO_SCREEN);
    BLOCKWRITE(TURBO_SCREEN,VISIBUF,128);
    CLOSE(TURBO_SCREEN);
                                    { And load it in again to prove }
                                     { it's OK }
      GOTOXY(13,12); WRITE('<<PRESS RETURN:>>');
      READLN;
      FILLCHAR(VISIBUF,SIZEOF(VISIBUF),CHR(0)); { This clears graphics }
                                                 { buffer }
      ASSIGN(TURBO_SCREEN,TURBO_NAME);
      RESET(TURBO_SCREEN);
      BLOCKREAD(TURBO_SCREEN,VISIBUF,128);  { BLOCKREAD the screen to }
                                            { buffer }
      CLOSE(TURBO_SCREEN);
      READLN;                               { Wait for (CR) }
      TEXTMODE                              { and exit }
    END
END.
```

Engineering Excellence

THE STATE OF THE ART IN DATA COMMUNICATIONS SOFTWARE

DESIGNED BY

MICROSTUF®

CROSSTALK IS A TRADEMARK OF MICROSTUF, INC., ATLANTA, GEORGIA

CROSSTALK IS AVAILABLE FOR MOST SMALL BUSINESS COMPUTERS

DATE 2/16/84

CIRCLE NO. 204 ON READER SERVICE CARD

Go ahead . . .

TAKE THE CREDIT!

★ **cash discount prices!**

★ **no annual or membership fee.**

★ You've made
CONROY-LAPOINTE
The World's Largest Computer
Mail-Order Firm!

★ So now we bring
you the convenience and
preferred status of being one of
OUR VERY OWN CHARGE CUSTOMERS!

★ Initial minimum purchase
is only $400. There's no minimum
purchase amount once your account is open.

★ **As a CONROY-LAPOINTE
charge customer, you'll benefit from
the convenience of credit priviledges . . .**

- **at cash discount prices!**
- **with no annual or membership fee.**
- **so you won't tie up personal or company cash.**
- **so you won't tie up other credit lines.**

Corporate customers:
★ you can use our credit ★
card like net 30 terms!

★ Send in the coupon
for your application

(800) 547-1289
TELEX 910 380 3980

DISKETTES

★ CONROY-LAPOINTE™ DISKETTES ★
We guarantee these top quality products with the Conroy-LaPointe name. 5 YEAR LIMITED WARRANTY. Also available w/o labels.

10 ea, DS/DD, (IBM, H/P) 48 Trk.	$ 15
100 ea, DS/DD, (IBM, H/P) 48 Trk.	$ 119
1000 ea, DS/DD, (IBM, H/P) 48 Trk.	$ 859

★ CONROY-LAPOINTE™ ★
IBM PRE-FORMATTED

10 ea, DS/DD, 48 Track (IBM-PC Pre-formatted)	$ 19
100 ea, DS/DD, 48 Track (IBM-PC Pre-formatted)	$ 149
1000 ea, DS/DD, 48 Track (IBM-PC Pre-formatted)	$ 959

		LIST	OUR
CDC,	10 ea, SS/DD, 40T (Apple, IBM)	$ 55	$ 21
	100 ea, SS/DD, 40T (Apple, IBM)	$ 550	$ 195
	10 ea, DS/DD, 40T (IBM, H/P)	$ 75	$ 32
	100 ea, DS/DD, 40T (IBM, H/P)	$ 750	$ 295
DYSAN,	10 ea, SS/DD (Apple, etc.)	$ 40	$ 27
	10 ea, DS/DD 48T(IBM,H/P)	$ 69	$ 35
MAXELL,	10 ea, SS/DD,MD1(Apple)	$ 55	$ 19
	10 ea, SS/DD, 3½" (MAC)	$ 60	$ 35
	10 ea, DS/DD, MD2 (IBM)	$ 75	$ 26
	10 ea, DS/QD Hi Dens(IBM-AT)	$ 77	$ 49
MEMOREX,	10 ea, SS/SD, 3½"(MAC)	$ 65	$ 35
VERBATIM,	10 ea,SS/DD,MD515-01,(Apple)	$ 49	$ 27
	10 ea, DS/DD, MD34 (IBM)	$ 84	$ 32

PRINTERS

DOT MATRIX:

	LIST	OUR
EPSON, RX80—100 cps	$ 269	CALL
RX80—F/T	$ 369	CALL
RX100—100 cps, 136 col., pin & fr.	$ 499	CALL
FX80—160 cps, 80 col.	$ 699	CALL
FX100—160 cps, 136 col.	$ 849	CALL
JX80—Color Printer, 160 cps	$ 399	CALL
LQ1500—200 & 67 cps	$1395	CALL
Tractor Feed for LQ1500	$ 70	CALL
MANNESMANN TALLY,		
Spirit—80 col., 80 cps	$ 399	$ 299
160—80 col., 160 cps	$ 798	$ 568
180—132 col., 160 cps	$1098	$ 778
OKIDATA, Okimate 20, Color, Hi Res	$ 268	$ 208
82A—80 col., 120 cps, para.	$ 349	$ 319
83A—132 col., 120 cps, para.	$ 749	$ 599
84—136 col, 200 cps, para.	$ 899	$ 729
92—80 col., 160 cps, para.	$ 499	$ 399
93—136 col., 160 cps, para.	$ 799	$ 649
2410P—Pacemark,350cps,para	$2995	$1995
QUADRAM, Quadjet, Inkjet Color Printer	$ 895	$ 795
STAR MIC., Gemini 10"X, 120cps, 18"	$ 499	$ 269
Gemini 15"X, 120cps	$ 549	$ 419
TOSHIBA, 1351—100 cps	$1895	$1375
1340—144 cps(DQ) & 54 cps(LQ) **NEW**	$ 995	$ 795
TTX, TTXpress, portable/handheld,40cps	$ 229	$ 129

LETTER QUALITY:

JUKI, 6300—40 cps, para	$ 995	$ 795
6100—18 cps, para, 3 pitch	$ 599	$ 449
TTX, 1014—13 cps, para/ser, p & fr, 3p.	$ 499	$ 365
1114—same as 1014 w/T&F, 2c & prop.	$ 599	$ 439

MONITORS

AMDEK, Color 300 Comp/Audio	$ 349	$ 249
Color 500 Comp/VCR/RGB/Audio	$ 525	$ 395
Color 600 Hi Res, RGB /Audio	$ 599	$ 459
Color 700 Ultra Hi Res, RGB	$ 749	$ 549
Color 710	$ 799	$ 599
300G, 12" Green	$ 179	$ 129
300G, 12" Amber	$ 199	$ 149
310A, 12" Amber, (IBM)	$ 230	$ 159
PRINCETON, HX-12, Hi Res, RGB	$ 795	$ 495
SR-12, Hi Res, RGB	$ 799	$ 599
Scan Doubler for SR-12	$ 249	$ 179
MAX-12, Amber (monochrome)	$ 249	$ 199
QUADRAM, Amberchrome, 12" Amber	$ 250	$ 165
Quadchrome 12" RGB Color	$ 695	$ 495
Quadchrome II, 14" RGB Color	$ 650	$ 450
Quadscreen 17" 968x512 w/cable, Hi Res	$1995	$1595
ZENITH, ZVM122, 12" Amber	$ 159	$ 95
ZVM123, 12" Green	$ 149	$ 89
ZVM124, 12" Amber	$ 200	$ 149
ZVM135, 12" Color	$ 599	$ 499

MODEMS

ANCHOR, Signalman Mark XII (IBM)	$ 399	$ 269
HAYES, 2400B External Modem (IBM)	$ 899	$ 699
Smartmodem 1200B (IBM)	$ 599	$ 409
Smartcom II Software (IBM)	$ 149	$ 99
Stack Chronograph (RS-232)	$ 249	$ 189
PROMETHEUS, ProModem 1200B (IBM)	$ 399	$ 289
QUADRAM, Quadmodem, Internal (IBM)	$ 595	$ 425
Quadmodem, External (IBM)	$ 695	$ 495
VENTEL, PC Half Card (IBM)	$ 549	$ 389
1200 Plus, External (IBM)	$ 499	$ 429
PC1200, Internal (IBM)	$ 499	$ 379

ACCESSORIES

Curtis, Diamond, 6 outlets, switched	$ 50	$ 29
Emerald, 6 outlets, 6' cord	$ 60	$ 35
Ruby, 6 outlets, 6' cord, filter	$ 90	$ 52
Sapphire, 3 outlets, w/filter	$ 80	$ 46
EPD, Lemon, 6 outlets/wall	$ 60	$ 29
Lime, 6 outlets/floor	$ 90	$ 45
Orange, 6 outlets/floor	$ 140	$ 60
Peach, 3 outlets/wall	$ 98	$ 39
Kiwi, 1 outlet/wall		CALL
NETWORX, Wiretree, 4 outlet, w/filt & surge	$ 70	$ 39
PERFECT DATA, Head Cleaning Kit	$ 16	$ 12
PROD TECH INT'L, Uninterruptable Power Supply		
200 Watts, PC200 for IBM-PC	$ 499	$ 399

THE WORLD'S LARGEST COMPUTER MAIL ORDER FIRM

#T15

CONROY-LA POINTE INC.

LOW PRICES TO PROFESSIONALS WHO KNOW WHAT THEY WANT AND KNOW HOW TO USE IT!

FOR YOUR IBM-PC, XT, AT or JR

COMPUTER SYSTEMS

— Call for Details —

256K IBM-PC

360K Disk Drives by CDC

90 Day Limited Warranty By Us

SM

COMPAQ Portable, 256K, 2 360K Disk Drives — CALL

SANYO 555-2, 256K, 2 320K Disk Drives — CALL

TeleVideo PC, 256K, 2 360K Disk Drives, 8088 Chip — CALL

ZENITH Z150, 256K, 2 320K Disk Drives, MS DOS 2.1, 8088 Chip, 2 S/P — CALL

HARD DISKS

Kits are completely engineered to work with DOS 2.0/2.1. Easy to install. Includes Hard Disk, Controller Card and Instructions.

	LIST	OUR
KAMERMAN, External 10 meg kit	$1295	$1095
Megaflight 100, Internal 10 meg kit	$ 895	$ 795
Masterflight, 20 meg, tape, surge prot.	$1595	$1150
MAYNARD, 10 meg kit, Int, w/Sandstar Cont.	$1595	$1150
QUADRAM, Quaddisks Int. w/controller, Full Line		CALL
RANA, External 10 meg w/controller	$1495	$1095
Internal 10 meg w/controller	$ 995	$ 795

DISK DRIVES

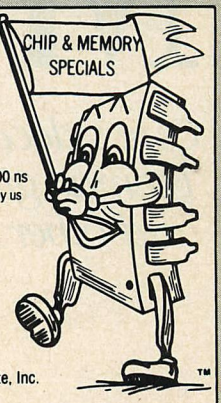
	LIST	OUR
CDC, Limited 30 day warranty; Call for quantity prices		
Full Ht, 320/360K, DS/DD	$ 529	$ 169
Half Ht, 320/360K, DS/DD	$ 395	$ 149
Half Ht. Dual Drive Installation Kit	$ 30	$ 16
MAYNARD, Controller Card w/para port	$ 300	$ 185
Controller Card w/serial port	$ 310	$ 195
Sandstar Cont Card (accepts 3 modules)	$ 265	$ 205
PERFECT DATA, Head Cleaning Kit	$ 16	$ 12

OTHER HARDWARE

	LIST	OUR
AST, SixPakPlus,64K,S/P/CC+S/W	$ 395	$ 245
SixPakPlus, 256K, S/P/CC+S/W	$ 695	$ 395
SixPakPlus, 384K, S/P/CC+S/W	$ 895	$ 465
Game Port for SixPak	$ 50	$ 39
Preview, Graphics Card w/para, 64K	$ 399	$ 299
Advantage™, Multif. Bd for AT	$ 595	$ 445
I/O Plus II, S/P/CC	$ 215	$ 150
I/O Plus II, S/P/CC/G	$ 265	$ 185
I/O Plus II, 2S/P/CC/G	$ 315	$ 215
MonoGraphPlus™P/CC(forLotus)	$ 495	$ 375
PCNet, Starter Kit, PC002	$1090	$ 795
PCNet, Circuit Board,PC001	$ 695	$ 365
ComboPlus Products IN STOCK		CALL
MegaPlus Products IN STOCK		CALL
CURTIS, UNI-I, Monitor tilt/swivel base	$ 50	$ 39
3-9 foot Keyboard Extension Cable	$ 40	$ 30
Vertical CPU "System Stand"	$ 25	$ 19
Monochrome Exten Cable Pair	$ 50	$ 35
HAUPPAGE (HC), 8087 Chip	$ 175	$ 149
8087 Math Pak (Chip & Softw)	$ 295	$ 235
8087 Software Pak	$ 180	$ 138
8087 Macro Pak	$ 245	$ 195
HAYES, Mach II Joystick	$ 45	$ 29
HERCULES, Color Card w/para.	$ 245	$ 169
Mono Graphics Card	$ 499	$ 329
KAMERMAN, External Power Supply	$ 395	$ 295
KENSINGTON, Masterpiece™	$ 140	$ 99
PC Saver™ Line Cord w/Filter	$ 50	$ 39
KEY TRONIC, KB5151, Std. keyboard	$ 255	$ 195
KB5150, Std. keyboard	$ 209	$ 159
KOALA, Speed Key System	$ 100	$ 63
Speed Key Tables w/software	$ 200	$ 139
Koala Pad™ w/PC Design	$ 150	$ 89
Programmer's Guide	$ 15	$ 14

OTHER HARDWARE

	LIST	OUR
MAYNARD, SAND STAR SERIES		
Multifunction (6) Card	$ 89	$ 79
Memory Card no RAM	$ 199	$ 169
Memory Card 256K	$ 495	$ 395
Floppy Cont. Card (WS1)(for 3 mod)	$ 265	$ 205
HardDisk I/F Module	$ 499	$ 399
HardDisk Cable	$ 30	$ 27
Serial Port Module	$ 95	$ 79
Para or Clock Cal. Module, ea.	$ 59	$ 49
Game Adapter Module	$ 49	$ 43
Memory Module, OK	$ 122	$ 99
Memory Module 256K	$ 422	$ 357
10 meg Hard Disk Kit & Cont Card	$1595	$1150
MICROSOFT, Mouse for PC	$ 195	$ 139
System Card, 64K	$ 395	$ 275
System Card, 256K	$ 625	$ 450
MOUSE SYSTEMS, PC Mouse & Paint	$ 295	$ 189
PARADISE, Modular Graphics Card	$ 395	$ 285
Parallel or Serial Port, ea.	$ 95	$ 69
PLANTRONICS,		
Color Bd & Colormagic, 16 color w/Para	$ 559	$ 395
Color Bd & Draftsman, 16 color w/Para	$ 559	$ 395
QUADRAM,		
Quadboard, no RAM, expand to 384K	$ 295	$ 225
Quadboard 64K, to 384K, S/P/CC	$ 395	$ 245
Quadboard 256K, to 384K, S/P/CC	$ 675	$ 395
Quadboard, 384K, S/P/CC/G	$ 795	$ 495
Quadboard II, no RAM, to 256K	$ 295	$ 215
Quadboard II, 64K, to 256K, 2S/CC	$ 395	$ 265
Quadboard II, 256K, 2S/CC	$ 595	$ 395
Quad 512 + 64K w/serial port	$ 325	$ 265
Quad 512 + 256K w/serial port	$ 550	$ 420
Quad 512 + 512K w/serial port	$ 895	$ 625
Quadcolor I, board, 4 colors	$ 295	$ 195
Upgrade Quadcolor I to II kit	$ 275	$ 199
Quadvue, board, Mono, S/P/CC	$ 345	$ 269
Quadchrome Monitor, 12" RGB Color	$ 795	$ 495
Quadchrome II Monitor, 14" RGB Color	$ 650	$ 450
Amberchrome Monitor, 12" Amber	$ 250	$ 165
Quad 3278	$1195	$1050
Quadnet VI	$2295	$1545
Quadnet IX	$1995	$1745
Quadlink	$ 495	$ 385
TG PRODUCTS, Joystick	$ 45	$ 29
TITAN, Accelerator PC (8086+ 128K)	$ 995	$ 750
WICO, Smartboard Keyboard	$ 400	$ 279



★ Memory Chip Kit ★

$35

9 Each, 4164, 200 ns
90 Day Warranty by us

Call for Larger Quantity Prices

★ ComX ★

EconoRAM™ 384K BOARD

$350

With Fastrak™ RAM Disk Emulator and Spooler Software
Fully Compatible, 1 Year Limited Warranty by ComX
Works on DOS 1.1, 2.0 or 2.1
Prices and availability subject to change. Call.

★★ FOR YOUR PC-JR ★★

	LIST	OUR
KEY TRONIC, KB5151 Jr. Keyboard	$ 255	$ 195
KOALA, Touch Tablet for Jr.	$ 125	$ 75
MOUSE SYSTEMS, Mouse for Jr.	$ 195	$ 125
MICROSOFT, 128K Booster w/Mouse	$ 495	$ 329
Serial Mouse	$ 195	$ 139
QUADRAM, Quadmemjr.(128K-512K,P/CC)	$ 275	$ 215
Quadjr. Expansion Chassis	$ 695	$ 540
TECMAR, Jr. Captain	$ 395	$ 345

SOFTWARE FOR YOUR IBM-PC, XT, AT or JR

BUSINESS

APPLIED SOFTWARE, VersaForm	$ 389	$ 249
ASHTON-TATE, Framework	$ 695	$ 349
dBase III	$ 695	$ 369
dBase II, (req. PC-DOS & 128K)	$ 495	$ 289
dBase II to III upgrade	$ 200	$ 119
ATI, Training Programs—Large Inventory	$ 75	$ 50
BPI, Job Cost Accounting	$ 795	$ 495
Gen'l Acctg, AR, AP or PR, each	$ 595	$ 375
BRODERBUND, Bank St. Writer(PC or Jr)	$ 80	$ 50
CDEX, Training Programs—Large Inventory	$ 70	$ 45
CONTINENTAL, Ultrafile (PC)	$ 195	$ 125
Tax Advantage (PC or Jr)	$ 70	$ 45
FCM (Filing, Cataloging, Mailing)(PC)	$ 125	$ 75
Property Management (PC)	$ 495	$ 295
DOW JONES, Investment Evaluator	$ 139	$ 99
Market Manager	$ 300	$ 159
Market Analyzer	$ 350	$ 219
Market Microscope	$ 350	$ 219
FOX & GELLER, dUtil(DOS or CP/M86)	$ 99	$ 65
Quickcode or dGraph, each	$ 295	$ 165
HARVARD, Total Project Manager	$ 495	$ 315
Harvard Project Manager	$ 395	$ 239
HAYDEN, Pie Writer	$ 200	$ 125
Pie Speller	$ 50	$ 30
HOWARDSOFT, Tax Preparer for '84	$ 295	$ 195
Real Estate Analyzer	$ 250	$ 170
HUMAN EDGE, Mind Prober (PC or Jr)	$ 50	$ 32
Communications Edge (PC)	$ 195	$ 119
Sales Edge	$ 250	$ 159
Management Edge	$ 250	$ 159
Negotiation Edge	$ 250	$ 159
IUS, EasyWriter II System	$ 350	$ 250
EasySpeller II	$ 85	$ 125
GL, AR, AP, OE or INV, each	$ 595	$ 375
Business System : GL+AR+AP	$1495	$ 995
KENSINGTON, Easy Link Mail Manager	$ 95	$ 59
LIFETREE, Volkswriter Deluxe	$ 395	$ 159
Volkswriter	$ 195	$ 105

BUSINESS

LIVING VIDEOTEXT, Think Tank	$ 195	$ 105
LOTUS, 1-2-3	$ 495	$ 309
Symphony	$ 695	$ 465
MDBS, Knowledgeman	$ 500	$ 300
MECA, Managing Your Money	$ 195	$ 125
MICROPRO, WordStar™ (PC)	$ 350	$ 189
WordStar™ (Jr) NEW	$ 195	$ 115
WordStar 2000	$ 495	$ 295
WordStar 2000 Plus	$ 595	$ 325
WordStar Professional Plus	$ 695	$ 395
WordStar Professional, 4 Pak	$ 495	$ 265
MailMerge, SpellStar or StarIndex, ea.	$ 99	$ 54
ProOptions Pak (MM/SS/SI)	$ 195	$ 105
InfoStar Plus (+ Starburst)	$ 595	$ 315
CorrectStar	$ 145	$ 77
MICRORIM, RBase Series 4000	$ 495	$ 269
Extended Report Writer	$ 150	$ 95
RBase Clout	$ 195	$ 125
MICROSOFT, Spell	$ 50	$ 32
Multiplan (PC or Jr)	$ 195	$ 125
Chart or Project, each	$ 250	$ 159
Word	$ 375	$ 235
Word with Mouse	$ 475	$ 289
MONOGRAM, Dollars & Sense w/Forecast	$ 180	$ 110
MULTIMATE, Multimate	$ 495	$ 295
OPEN SYS, GL,AR,AP,PR,INV or PO,ea.	$ 695	$ 429
PEACHTREE, Back to Basics GL	$ 295	$ 175
Peach Pak	$ 395	$ 239
Peach Text 5000	$ 395	$ 239
QUADRAM, Tax Strategy	$ 395	$ 295
Investment Strategy	$ 395	$ 295
QUE, Using 1-2-3	$ 15	$ 12
1-2-3 for Business	$ 15	$ 12
Using Symphony	$ 20	$ 15
SAMNA, Word Plus	$ 295	$ 185
SATELLITE, WordPerfect (PC)	$ 495	$ 235
WordPerfect (Jr)	$ 69	$ 49
SOFTW ARTS, TK Solver(specif DOS)	$ 399	$ 269
SOFTWARE INT'L, Open Access	$ 695	$ 395

BUSINESS

SOFTWARE PUBL., PFS:Report	$ 125	$ 79
FS:File	$ 140	$ 89
PFS:Write	$ 140	$ 89
PFS:Graph	$ 140	$ 89
PFS:Plan	$ 140	$ 89
PFS:Proof or PFS:Access, each	$ 95	$ 59
SORCIM, SuperCalc III	$ 395	$ 245
STONEWARE, Advanced DB Master	$ 595	$ 395
SUMMA, Trader's Forecast	$ 250	$ 159
Trader's Data Manager	$ 200	$ 129
Trader's Accountant	$ 350	$ 219
Complete System	$ 700	$ 445
THORN/EMI, Perfect Writer (PC)	$ 349	$ 179
Perfect Combo (Writer & Speller) (PC)	$ 399	$ 199
Perfect Combo (Jr) (Write/Spell/Thesaus	$ 139	$ 89
VISICORP, VisiCalc 4	$ 250	$ 159
WARNER, Desk Organizer (PC or Jr)	$ 195	$ 125

UTILITIES

BORLAND, Sidekick (PC or Jr)	$ 55	$ 35
Sidekick (Copiable) (PC or Jr)	$ 85	$ 55
Turbo Pascal (PC or Jr)	$ 55	$ 35
Toolbox (PC)	$ 55	$ 35
CENTRAL POINT, Copy II PC	$ 40	$ 30
COMX, Fastrak™, RAM/Disk emulator & printer spooler. For any PC/DOS vers. or RAM Card.Menu Driven	$ 100	$ 59
DIGITAL RES., CP/M-86™ (PC/XT)	$ 80	$ 39
CBASIC 86™ (CP/M-86)	$ 200	$ 135
CBASIC Compiler(CP/M-86 or PCDOS, ea)	$ 600	$ 395
Concurrent CP/M-86™ w/windows	$ 835	$ 225
PL/1 (PC DOS)	$ 750	$ 495
Speed Prog. Pkg. (CP/M-86)	$ 200	$ 135
DR LOGO-86 (CP/M-86)	$ 100	$ 69
FUNK SOFTWARE, Sideways	$ 60	$ 40
HAYES, Smartcom II (Data Comm.)	$ 149	$ 99
LIFEBOAT, Lattice C	$ 500	$ 295
MICROSTUF, Crosstalk XVI (PC or Jr)	$ 195	$ 129

UTILITIES

MICROSOFT, Macro Assembler	$ 100	$ 69
BASIC Compiler	$ 395	$ 259
Business BASIC Compiler	$ 600	$ 300
C Compiler	$ 395	$ 259
COBOL Compiler	$ 700	$ 459
FORTRAN Compiler	$ 350	$ 229
PASCAL Compiler	$ 300	$ 199
MOUSE SYSTEMS, PC Paint	$ 99	$ 69
NORTON, Utilities 2.0 (14 programs)	$ 80	$ 54
OPEN SYSTEMS, BASIC Interpreter	$ 195	$ 145
ROSESOFT, Prokey	$ 130	$ 79
WESTERN UNION, Easy Link Mail Mngr	$ 95	$ 65

HOME & EDUCATIONAL

ARMONK, Executive Suite	$ 40	$ 27
CONTINENTAL, Home Accountant (Jr)	$ 75	$ 59
Home Accountant Plus (PC)	$ 150	$ 90
KOALA, Graphics Exhibitor (Jr)	$ 40	$ 25
MONOGRAM, Dollars & $ense w/forecast	$ 165	$ 110
SCARBOROUGH, MasterType(PC or Jr)	$ 50	$ 32
SIMON & SCHUSTER, Typing Tutor III	$ 50	$ 33

PLUS: **BPI, CBS, Comprehensive, Davidson, Dow Jones, Harcourt, PBL Corp.**

RECREATIONAL

BLUECHIP, Millionaire, Barron, Tycoon, ea.	$ 60	$ 39
HAYDEN, Sargon III (Chess)	$ 50	$ 34
MICROSOFT, Flight Simulator (PC or Jr)	$ 50	$ 33
ORIGIN, Ultima III (PC or Jr)	$ 60	$ 39
PROFESSIONAL, Trivia Fever (PC or Jr)	$ 40	$ 25

PLUS: **Atari, Broderbund, Electronic Arts, Epyx, Infocom, Insoft, Spinnaker, Sub Logic**

CASH-n-CARRY COMPUTER STORES, INC.

Over-the-counter sales only. Open Monday through Saturday, 10:00 to 6:00.

SAN FRANCISCO — NEW STORE! 550 Washington Street (at Montgomery, opposite the Pyramid). Interstate 80, to Highway 480; take Washington Street Exit. **CALL (415) 982-6212.**

PORTLAND, OREGON — At Park 217, Tigard at intersection of Highways 217 and 99W. **CALL (503) 620-5595.**

SEATTLE, WASHINGTON — 3540 128th Ave. SE, Bellevue, WA 98006. In Loehmann's Plaza near Factoria Square, South East of Highway 405 & 90 and at South East 36th and Richards. **CALL 641-4736.**

MEMBER DIRECT MARKETING ASSOCIATION

OUR REFERENCES:

We have been in computers and electronics since 1958, a computer dealer since 1978 and in computer mail order since 1980. Banks: 1st Interstate Bank, (503) 643-4678. We belong to the Chamber of Commerce (503) 2289411, and Direct Marketing Association, or call Dunn and Bradstreet if you are a subscriber.

MasterCard VISA American Express Card CONROY-LAPOINTE CARD

NO SALES TAX

National Order Desk **TOLL FREE (800) 547-1289**
OREGON ONLY (800) 451-5151

Foreign & Portland (503)620-9877

HOT LINE
Information on your order
(503) 620-9878
WEEKDAYS ONLY

ORDER DESK HOURS
Mon-Fri—6AM to 6PM PST
Saturday—10AM to 4PM PST
(6AM here is 9AM in New York)

CONROY-LAPOINTE CREDIT CARD

Send me a Conroy-LaPointe Credit Card application form. There's **no 3% charge** on Conroy-LaPointe Credit Card purchases. Minimum initial purchase is $400.

NAME

ADDRESS

CITY STATE **# T15** ZIP

MAIL TO: 12060 SW Garden Place, Portland, OR 97223

ORDERING INFO & TERMS:

MAIL TO: 12060 SW Garden Place, Portland, OR 97223 —— Include your telephone number; double check your figures for Shipping, Insurance and Handling (SIH). All items usually in stock. Compatability is customer's responsibility. **NO C.O.D.** Cashiers checks, money orders, Fortune 1000 checks and government checks honored immediately. Personal and other company checks—allow 20 days to clear. Prices reflect **3% cash & Conroy-LaPointe Credit Card discount,** so ADD 3% to above prices for VISA, MasterCard or American Express. Add **SIH CHARGES:** U.S. Mainland, 3% ($5 minimum) for standard UPS ground; UPS Blue, 6% ($10 minimum); for U.S. Postal, APO or FPO, 12% ($15 minimum). Hawaii—UPS Blue. For Alaska or Canada—UPS in some areas only, all others Postal—call or write, or specify Postal. Foreign orders except Canada, 18% ($25 minimum). Monitors by Postal or to foreign countries, 30% ($50 minimum). Orders received with insufficient SIH charges will be refunded. All prices, availability and specifications subject to errors or change without notice, so call to verify. All goods are new, include warranty and are guaranteed to work. Due to our low prices and our assurance that you will get new, unused products—**ALL SALES ARE FINAL.** We do not guarantee compatibility. Call before returning goods for repair or replacement. **ORDER DESK HOURS—6AM to 6PM PST, Monday through Friday, Saturday 10 to 4.** (6AM here is 9AM in New York.) **EconoRAM™, Fastrak™, and Generik™ are trademarks of ComX Corporation. PC MasterCard™ and SoftRAM™ are trademarks of Magnum Computer.**

Memory Sidecars

One sidecar from IBM and two from Tecmar give PCjr increased memory and other useful features.

AUGIE HANSEN

Too much has been said and written about the original PC*jr* keyboard, and as Popeye was fond of saying, "Too much is enough." The *real* problem with the PC*jr* (the deficiency that kept me from buying one at first) was the seemingly contrived limitation on the amount of user memory available. This made it difficult or impossible to run many of the programs necessary to my work. Since the principal tasks for which I use my computer involve programming and writing, I use some large programs (compilers and word processors) that just cannot run on a one-disk, RAM-starved system. Of course, the limited free memory made it impractical to set up a RAM disk; most copy-protected programs do not permit a RAM disk anyway.

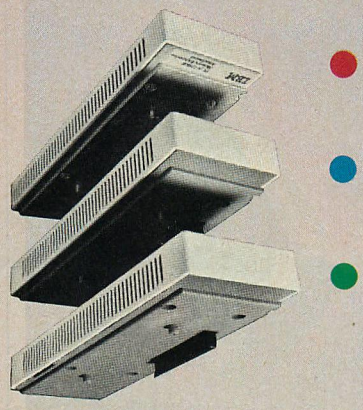
The PC*jr* has two basic configurations: the entry version, with 64KB RAM and no disk, and the enhanced version that sports both a double-sided disk and 128KB RAM. Display memory is in main RAM instead of on a separate adapter card with its own memory space allocation. This is a major hurdle that many programs cannot clear. In addition to stealing between 16KB and 32KB of RAM for exclusive use by the video system, this puts a barrier between the available program memory (about 88KB) and any additional memory that may be placed above the 128KB boundary of the enhanced system.

The PC*jr* has experienced the same sort of post-release influx of enhancement products as did many of its siblings in the IBM family. After-market suppliers for *jr* include IBM itself and Tecmar, a company that has provided a range of useful enhancement products, including memory expansion modules.

When IBM introduced the new keyboard for the PC*jr*, it also got into the expansion game by announcing a 128KB memory expansion attachment and a power expansion attachment, plus a set of programs collectively called "PC*jr* Memory Options," that can put nearly a half-megabyte of memory at a program's disposal. The add-on memory is housed in a narrow box that has been dubbed a *sidecar*. It attaches very neatly to the right side of *jr* where the bus-expansion connector is located. Increased memory and other features provided by sidecars make them highly desirable additions to a PC*jr* system.

THE VIDEO ROADBLOCK

The PC*jr* has a built-in video color/graphics subsystem. The video interface of the PC*jr*, while completely different from those used in the PC and PC/XT, is



compatible with them because of some tricks used in the way display memory is addressed. Figure 1, a memory map of the enhanced PC*jr*, shows that display memory is part of main RAM. (In the entry-level PC*jr*, the top of memory is at 64KB instead of at 128KB.) The map shows low memory at the top and higher memory addresses at progressively lower positions. As illustrated in figure 2, the memory within the unexpanded configuration is organized as "pages" of 16KB each.

The CRT/Processor Page Register, a write-only register at address 3DFH (figure 3), controls the assignment of CRT and processor pages for video use. The meanings of some bits vary with the video mode and the amount of installed memory. In a 64KB system, bits 2 and 5 do not participate in selecting pages. Thus, four pages are available in the 64KB system and eight in the 128KB version. Since graphics modes use 32KB blocks of RAM, bit 0 is ignored when graphics modes are selected.

To maintain compatibility with programs written for earlier members of the IBM PC family, memory reads and writes to the locations at segment B800H are automatically mapped to the video memory page selected by the processor page bits. No speed penalty is exacted on programs using this feature. The CRT page selection determines which page or pages are mapped to the display driver. It is possible, therefore, to write to one video page while displaying another. Simply changing the selected CRT page permits instantaneous page flipping, a very useful feature for high-speed animation.

This animation technique is used extensively in video games and even in some business applications. It is applicable to the color/graphics adapter directly, and in an indirect way can be applied to the IBM monochrome display adapter as well. In computer animation, the impression of movement can be created on the screen through a series of erasures and redrawings of objects. If done in realtime on the viewing screen, such techniques often result in visible flicker and image distortions. With multiple video pages (two is enough), a programmer can let the user view a visual page that is static while erasing, relocating, and redrawing figures on another active page of display memory.

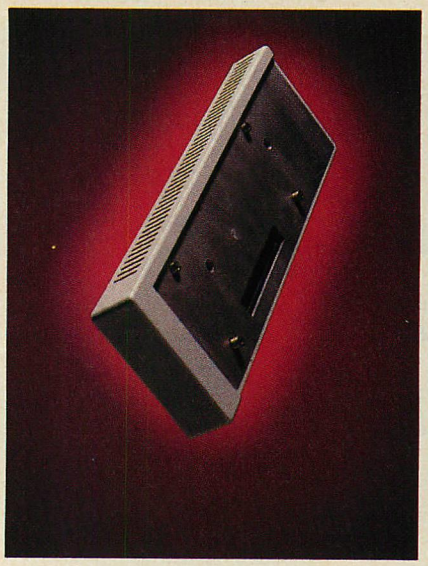
The pages are then swapped so the user is looking at the most recently updated page, and the process repeats with new data. The swap is achieved by setting pointers to the starting locations of the video pages. One of the pointers tells the display adapter where the data for refreshing the screen image (a frame) begin, and the other tells the program where to store the updated display data for the next frame. In BASIC, the SCREEN statement takes care of nearly all of the grunt work.

```
APAGE = 0: VPAGE = 1: SCREEN 0, 1,
  APAGE, VPAGE
```

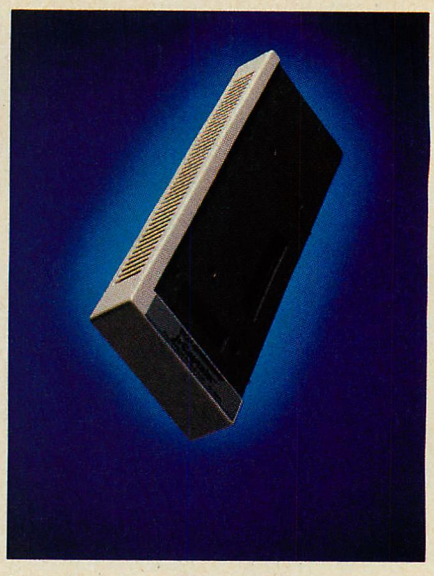
sets the color text mode, with the user looking at video page 1 and the program writing to video page 0. A sequence of PRINT statements can be used to place text and character graphics on the active page, then a simple

```
SWAP VPAGE, APAGE: SCREEN , , APAGE,
  VPAGE
```

will reverse the roles of the video pages. Used in a tight loop, this simple

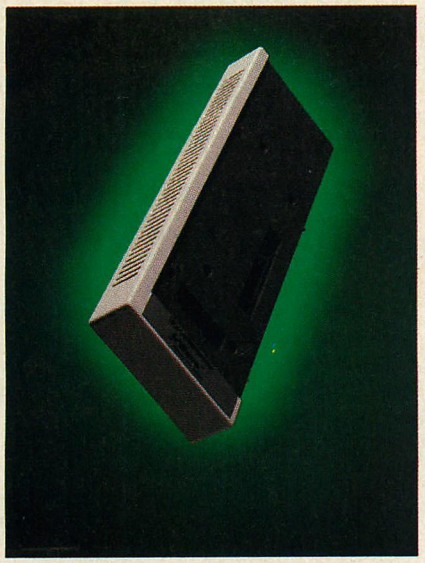


The IBM memory expansion sidecar adds 128KB of RAM. Three units can be attached.



In addition to 128KB, the Tecmar *jr*Captain provides a clock and parallel port.

The installed sidecar integrates well with the PC*jr* chassis.



The Tecmar *jr*Wave expansion sidecar can add a maximum of either 256KB or 640KB.

technique can produce some very good animation effects.

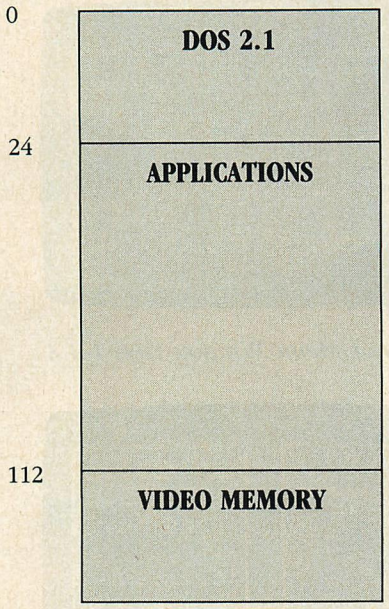
So, what does all of this mean? First, display memory cannot be placed above the 128KB boundary. The page-select register does not have enough bits to keep track of assignments outside of that range. Second, most applications need contiguous blocks of memory to work within—they cannot tolerate work spaces that are broken into chunks. Programs can be written to handle fragmented memory allocations, but this is not the case with most existing programs that were written for the PC and PC/XT. The software provided by IBM and Tecmar solves that problem and allows applications programs to access all available memory in the main system unit and in expansion units.

Both IBM and Tecmar produce well-engineered and manufactured packages. The IBM unit takes the usual competent, conservative approach. The Tecmar products have a decided edge in the flexibility afforded by their designs and the extra features available.

The IBM expansion unit holds 128KB of RAM. One unit can be powered directly off the main PC*jr* expansion bus. A power-expansion attachment that provides 20 watts is needed to permit use of any additional attachments. Another two units may be added to bring memory up to 512KB.

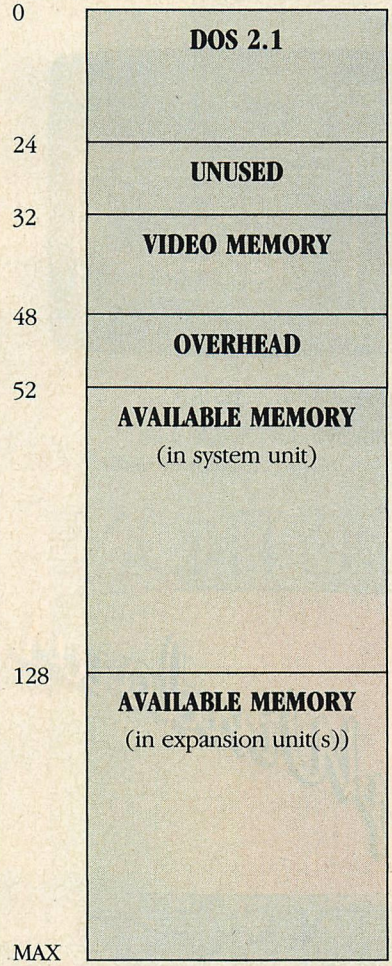
Tecmar's *jr*Captain includes a clock function and a parallel printer port along with 128KB of memory. Another Tecmar product, *jr*Wave, permits the use of either 16KB or 256KB memory chips in the first two banks and 64KB chips in the remaining two banks for a maximum of either 256KB or 640KB. Each sidecar has its own external power

FIGURE 1: *Memory Map for an Enhanced PCjr*



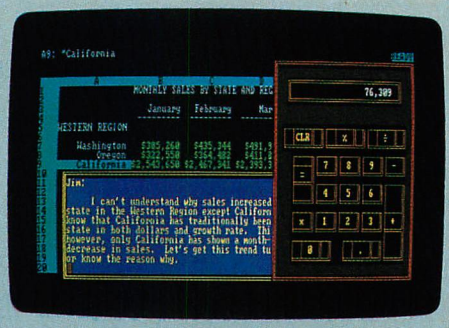
In the memory map for an enhanced (but unextended) PC*jr*, the video memory is in the highest 16KB of RAM, rather than on an adapter card with its own memory allocation.

FIGURE 2: *CRT/Processor Pages*

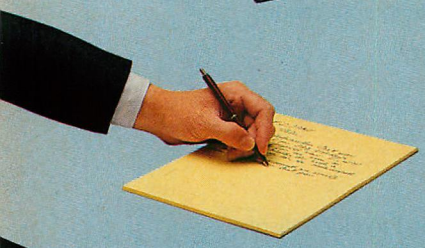


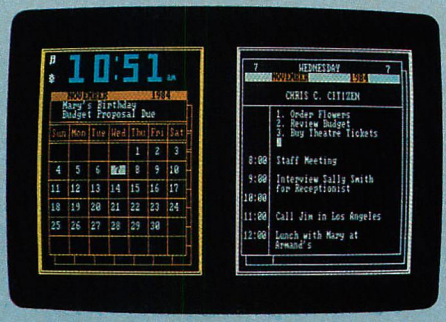
Each CRT/processor page corresponds to 16KB of memory. Speedy animation effects can be obtained by writing to one page while displaying another.

Replace this. With this. Instantly!

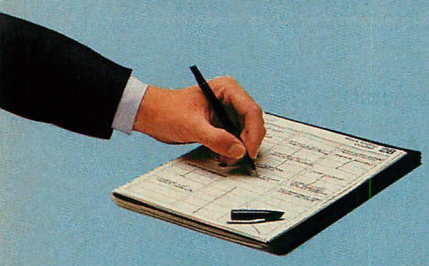


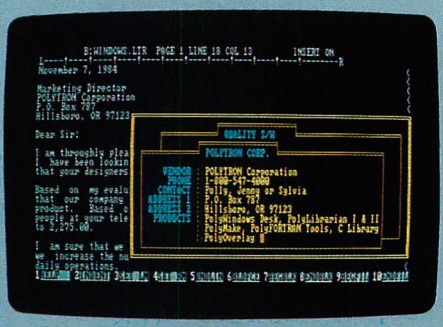
Calculator, Memo Window Over 1-2-3





Alarm Clock, Calendar, Appt. Book





Filedex Over WordStar

PolyWindows works with
a monochrome or color monitor

Desktop productivity in a single keystroke even while running another program!

Finally, a software package that makes your IBM PC or true compatible as useful as you thought it should be.

PolyWindows Desk™ lets you instantly call up essential desktop tools without having to exit the program you are running or swap disks. Another keystroke takes you back to your program right where you left off.

Your numeric keypad can now become a full-featured calculator or printing calculator.

You can create your own small scale data base and replace your manual Rolodex® with a Filedex system that has an unlimited number of "cards."

To help you manage your day, week, month and long-term projects, PolyWindows Desk has a monthly calendar, an alarm clock and daily appointment books.

And for the memos, letters and "to do" lists you write every day you can call up one or more "memo windows" with a built-in full-featured editor.

You can assign functions to specific keys using PolyKey,™ a keyboard enhancement program that rivals all the high-priced keyboard utilities. This feature alone justifies the price of PolyWindows Desk.

You can designate the size of each window and, with a color monitor, change the colors of windows and information within the windows. Pull-down menus give you the ease of a Macintosh™ without a mouse. You can even call up all these tools at once using PolyWindows, the most flexible, friendly windows yet! Plus, PolyWindows lets you include custom functions and is expandable for future utility modules. PolyWindows even includes PolyGames for a quick diversion.

At $49.95, PolyWindows Desk outperforms the competition and gives you the office of the future, today!

To order with VISA or MasterCard or request a complete information packet call toll-free. (Requires MS-DOS, PC-DOS 2.0, or higher).

The New Standard For Desktop Utilities

Only $49.95

Shipping $4

POLYTRON

PolyWindows™

DESK

PolyWindows was written by Thomas A. Crispin

1-800-547-4000

(Ask for Dept. No. 307)

Oregon & Foreign Orders Call
503-684-3000

Send Checks to:
Polytron Corporation, Dept. 307
P.O. Box 787
Hillsboro, OR 97123

Dealer/OEM inquiries welcome

pack, so there is no power drain on the PC*jr* expansion bus; this does, however, increase the number of electrical cords that must be plugged into the power mains. (I added a second power distribution strip to my system. It is daisy-chained off the first strip so all units are powered up at the same time under the control of one master switch.)

The physical and logical installation of the expansion units is described fully in the documentation provided by the manufacturers. The Tecmar *jr*Captain and *jr*Wave are both assigned the same starting address, so they cannot be used in tandem. If it is necessary to add more memory to a system equipped with a *jr*Captain, for example, Tecmar makes other expansion units containing banks of memory that may be assigned arbitrary starting addresses. Thus, the add-on memory may be merged with existing memory or may be broken into noncontiguous blocks, permitting users to hide blocks of memory from programs to meet special needs.

As explained earlier, the expansion memory is of no value to many programs because they cannot tell it is there. To get around this limitation, both IBM and Tecmar offer special software that rearranges the allocation of memory to create contiguous blocks of user memory for applications programs and in-memory disks that simulate the presence of a second disk drive.

The IBM memory options package comprises several programs. The programs are DOS device drivers that attach themselves to DOS 2.1 at the low end of memory. PCJRMEM.COM reallocates video memory and RAMDISK.COM handles the RAM-disk set-up. The drivers are named in a CONFIG.SYS file read by DOS at boot time. Menu-driven installation makes creating systems and applications program disks a relatively painless task.

DOS 2.1 is always placed at the bottom of memory. With no option switches set, PCJRMEM creates a 16KB video area and defines the start of user memory at 52KB. One of two switches may be specified (on the command line in CONFIG.SYS) to cause larger video memory allocations to permit use of PC*jr* high-resolution graphics modes. Table 1 summarizes the modes of operation, and figure 4 shows the resulting memory maps. Up to 96KB may be earmarked for the video buffer, permitting high-resolution graphics pages. But even if these pages are not needed, allocating 96KB to the video buffer forces the user program to run from the expansion memory. This is a key advantage since the video refresh function steals cycles from the main system memory only; thus, PC performance is possible with PC*jr* programs that execute from the expansion unit memory.

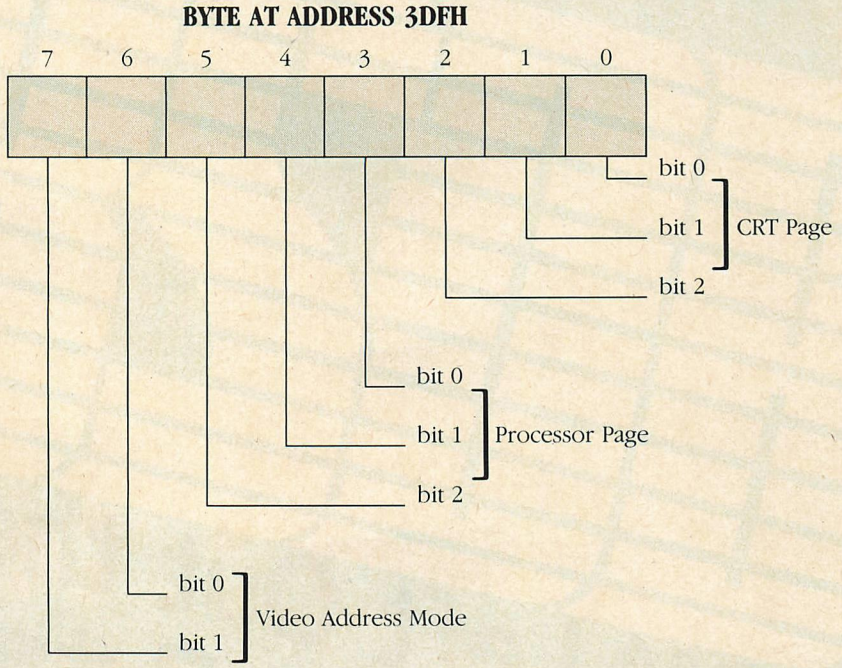
That PC-like performance will accrue only to programs that are not I/O bound. The Sieve of Eratosthenes (10 iterations, 1,899 primes) compiled by the Lattice C compiler runs on a PC/XT in about 9 seconds. On a PC*jr*, using a normal DOS configuration with the program executing in main memory, it takes 23 seconds. Loading the PCJRMEM driver in full compatibility mode places the executing program in expansion memory and it runs in 9 seconds, just like on its big brother.

The situation changes dramatically for programs that are disk- and display-intensive, however. Using the Lattice C compiler again, a small program (cat.c) was compiled and linked on the PC*jr* and on an XT under varying conditions. The cat.exe output file was ready to run in 122 seconds under the standard PC*jr* configuration and in 89 seconds, a 27-percent speed improvement, under the compatible arrangement. Contrast that with 57 seconds on an XT using a floppy disk and a mere 25 seconds on the fixed disk. Almost no screen output is required, so the primary factors at work here are the stolen CPU cycles and the lack of DMA during disk reading and writing on the PC*jr*, and the considerable differences in speed between floppy-disk and fixed-disk accesses on the XT. Even when executing programs in expansion memory, the PC*jr* may be much slower than a PC when disk interactions are required.

My Chrome Ranger game is an example of a program that is split evenly between calculations (maze positions, collision detection, scoring, etc.) and video display accesses (erasing and redrawing various animated figures). In the process of revising the game to make it work on the PC*jr* at the same speed as on a PC, I had to know how much slower it ran on the former machine. A reasonable measure of speed in this case is the time it takes to move the Chrome Ranger from one corner of the maze to another. The time on a PC, on either the color or standard monochrome display, is 6 seconds. On a PC*jr*, that increased by nearly 200 percent to 17 seconds. In the compatible mode, with all calculations being done in expansion memory, it still took a long time—roughly 14 seconds—to traverse the same path.

The conclusion drawn from these test results is that programs that are not interacting heavily with users and disks can run as fast on a PC*jr* in compatibility mode as on a PC. Spreadsheets and some other business tools will be the primary beneficiaries, except when they are doing file updates and drawing graphics displays. Most database programs will be slower on a PC*jr*, even with the expansion memory.

FIGURE 3: *CRT/Processor Page Register*



The CRT/processor page register controls the assignment of pages for video use. The register is write-only and determines which pages are mapped to the display driver.

IBM Left Out the Speed . . . Hauppauge Puts It Back!

From the inventors of the 87 Chip-in-a-Carrier Comes . . .

287FAST/5

The Highest Performance Math Coprocessor Module for the PC/AT

In an IBM PC/AT, the ordinary 80287 Math Coprocessor Chip runs at 4MHz. Now Hauppauge, the inventors of the 87 Chip-in-a-Carrier, bring you the 287 FAST/5, a module that delivers full 5MHz of Math Coprocessor power to your IBM PC/AT.

- **25% faster Math for the PC/AT!** Only the Hauppauge 287 FAST/5 can run at 5MHz in a PC/AT.
- **Easy installation!** The 287 FAST/5 module is carrier mounted for easy installation and immunity from bent pins.
- **Low price!** The 287 FAST/5 is no more expensive than most ordinary 80287 chips.

287 FAST/5 **$345.**
80287 Math Coprocessor module for full speed 5MHz operation in the PC/AT. Includes diagrammed installation instructions and diagnostic diskette.

287 Chip **$295.**
80287 Math Coprocessor chip. 5MHz chip, but only runs at 4MHz in an IBM PC/AT. Carrier mounted for easy installation. Includes installation instructions and diagnostic diskette.

87 Chip **$175.**
8087 Math Coprocessor chip carrier mounted for easy installation in the PC, PC/XT and PC compatibles. Includes diagrammed installation instructions and diagnostic diskette.

87 Software Pak **$180.**
Math interface libraries for the IBM BASIC, PASCAL and FORTRAN Compilers, also for the IBM MACRO Assembler. Plus, a set of high speed Matrix, manipulation routines. Includes complete source code, a one year Software Update Service, and the BEST book on 8087 programming, 8087 Applications and Programming by Richard Startz.

(The 87 Software pak is only $120. when purchased with either the 287 FAST/5 or the 87 Chip!)

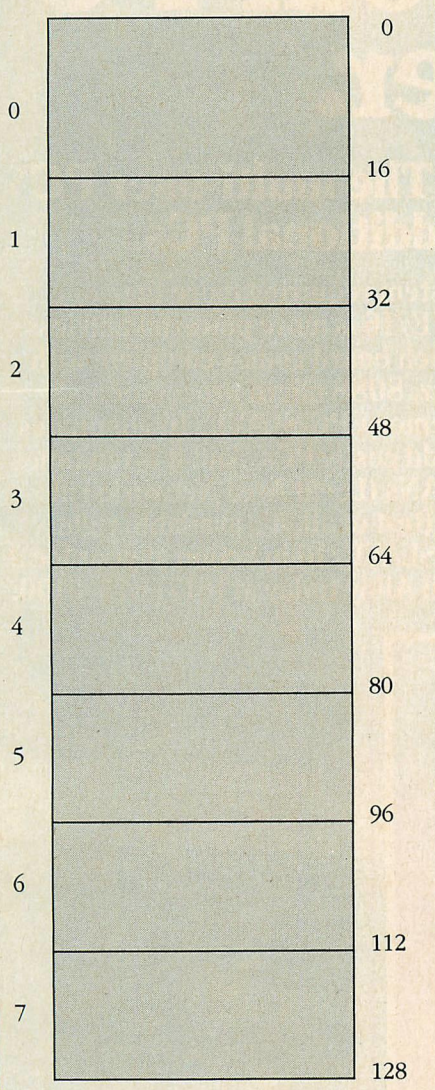
Hauppauge Computer Works Inc. **HCW**

358 Veterans Memorial Highway, Suite MS1
Commack, New York 11725
516-360-3827

AVAILABLE FROM YOUR LOCAL COMPUTER DEALER. IBM PC/AT, PC and PC/XT are trademarks of International Business Machines Corporations.

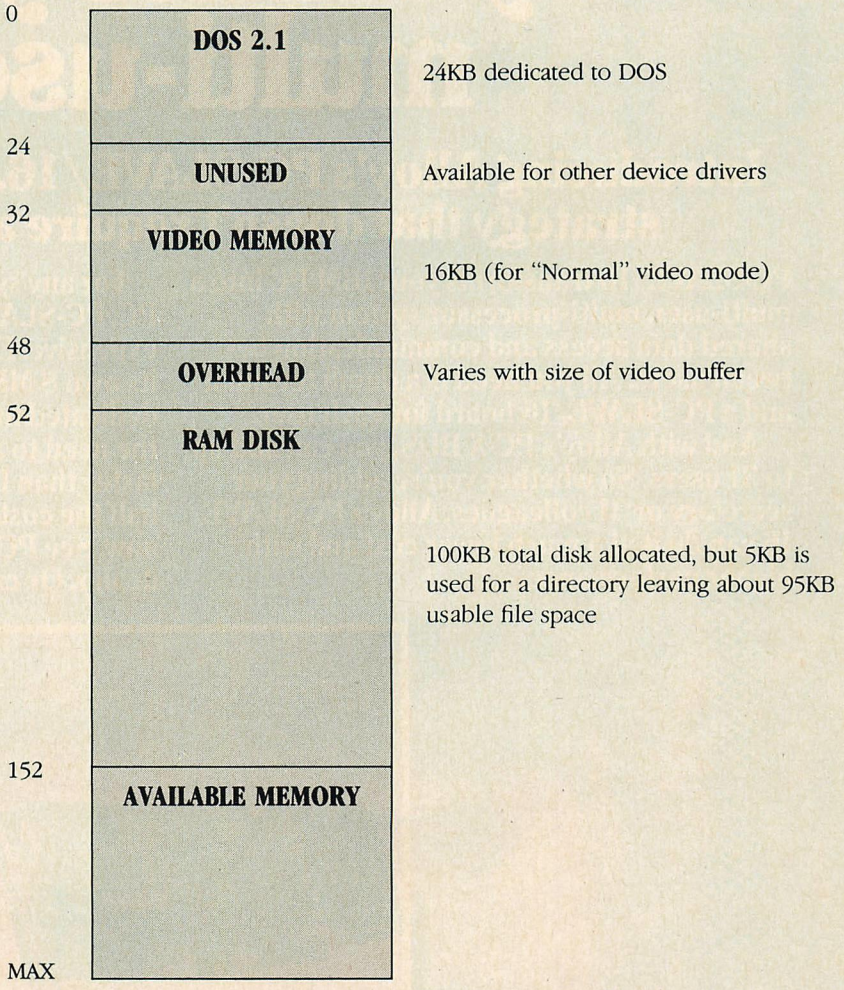
CIRCLE NO. 134 ON READER SERVICE CARD

FIGURE 4: *Memory Map for an Enhanced, Extended PCjr*



Device drivers remap the video memory to follow DOS in low memory.

FIGURE 5: *Memory Map for Video Buffer of 16 Kilobytes and RAM Disk of 100 Kilobytes*



This memory map shows allocations for a video buffer of 16KB followed by a 100KB RAM disk. The RAM disk may be combined with any of three video allocation schemes.

It is perhaps advisable to run with both memory expansion and RAM disk installed. The RAM disk may be combined with any of the three video allocations. Figure 5 shows memory allocations for a RAM disk of 100KB and the normal video buffer of 16KB.

Tecmar provides companion software with *jr*Captain (though not with *jr*Wave). A program named CONPCJR.EXE is executed from the AUTOEXEC.BAT file. It is not a device driver. It configures memory to a user's specifications, allocating 32KB by default for a video buffer. Pages 1 and 2 are reserved for DOS, so they may not be selected. CONPCJR takes combinations of options using a dash as a switch character, regardless of the value of SWITCHAR in the DOS environment. A '–A' option tells the program to set up 16KB for a text-only video buffer, saving 16KB for program use. The '–P' option takes a number in the range of 2 to 7 to specify which page to use for text.

When selecting a location for the video buffer, the user specifies only the 16KB text page. A full 32KB is set aside for the buffer; the program automatically aligns the buffer so that it will begin on a 32KB boundary—a requirement imposed by the way PC*jr* processes graphics display information.

MEMDISK.EXE, allocates a RAM disk of a specified size; by default, it allocates all memory up to 360KB.

The resulting memory maps for Tecmar's companion software resemble those that describe IBM's PC*jr* Memory Options programs. The absolute addresses change slightly, but the blocks of memory are managed in a similar way. A careful choice of pages must be made with the Tecmar CONPCJR program to avoid leaving a large gap in the memory map betwen the top of DOS and the base of the video buffer.

Purchasers of *jr*Captain are treated to a bonus: a set of programs called the Treasure Chest. Of the 24 programs in the package, most are useful, some are gems, and only a few have the ring of a wooden nickel.

Here is a brief look at each of the programs in the package. They work on all members of the PC family, but require a Tecmar multifunction board.

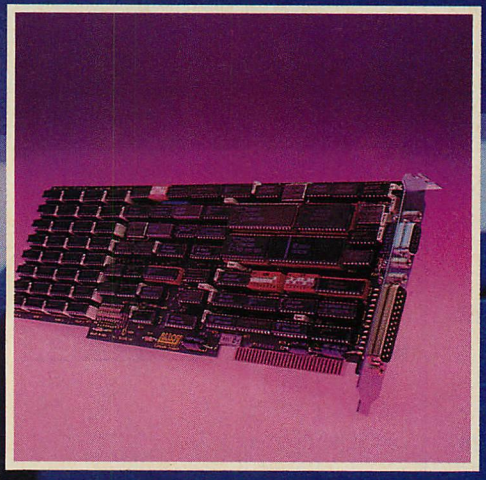
Alloy makes your PC multi-user.

Introducing Alloy's PC-SLAVE/16. The multi-user PC strategy that doesn't require additional PCs.

PC-SLAVE/16 transforms your IBM PC into a multi-user, multi-processor system with shared data access and communication between users. All you need is a dumb terminal and a PC-SLAVE/16 board for each user.

And with Alloy's PC-X BUS, you can add up to 31 users. So whether you're adding your secretary or 31 colleagues, Alloy's PC-SLAVE/16 can do it at a much lower cost and with higher performance.

For any system requiring more than one PC, PC-SLAVE/16 is more powerful and less expensive. Each board comes complete with its own 8 MHz, 8088 micro processor and dedicated memory. PC-SLAVE/16 eliminates the need to add costly PCs. And it is ten times faster than LANs (Local Area Networks) at half the cost. Best of all, it's available right now. See PC-SLAVE/16 in action at your local dealer. Or call Alloy today at (617) 875-6100.



ALLOY

ALLOY
Computer Products, Inc.

Alloy Computer Products, Inc., 100 Pennsylvania Avenue, Framingham, Massachusetts 01701. (617) 875-6100, TWX: 710-346-0394
In Europe: Alloy Computer Products (Europe) Ltd., Cirencester, Gloucestershire, England. Tel: 0285-68709, Tlx: 43340

CIRCLE NO. 111 ON READER SERVICE CARD

BANNER produces a magnified version of a message on the screen, on a printer, or in a disk file. The characters are Gothic-looking, in one of two selectable sizes, using either the default X character or one specified by an option flag to the command. The banner prints letters turned sideways and progressing down the page.

CAL creates a stock calendar for specified month or year and reports the current date and time by default. Information about holidays is stored in a separate file called CAL.DAT. It may be edited to include important dates.

CALC is a programmer's style calculator with memory. It may be kept resident in memory or be called in as needed. An option allows the use of color on systems equipped for it.

CHECK is yet another checkbook balancing program. It includes the usual credit and debit transactions and a reconciliation function to assist the user in balancing an account that has outstanding credits and debits.

COPRINT is a utility program that provides a limited amount of control over one or more printers through line printer spooler. A related program, PRINTER, is used to establish print spoolers in memory. The options to the spooler program control where in memory a spooler is loaded, the size of the buffer, and what type of printer is attached to each available port. It also initializes serial printer ports. On a standard PC, spoolers can be in a noncontiguous blocks of memory above the value set into the memory size switches. Because of the way the PC/XT and PC*jr* size memory dynamically, options to control placement are not permitted.

CRON, TICK, and REMIND work together to provide times reminders, an on-screen clock, and a timing function.

ENCODE and DECODE provide a reasonable degree of file security by scrambling and unscrambling the contents. A single key typed by the user controls the actual encoding and decoding sequences.

DOSTIME, SETTIME, and TIMTEST are used to control and access the Tecmar clock and to tell DOS what time the clock thinks it is.

FORM can be used with data files created by other Treasure Chest programs to create personalized mailings by merging the contents of a text file with data from an address list or "Rolodex" data file.

MEMO is a menu-driven editor program used to create simple text files. The files are often processed by other programs in this set to produce output such as sorted lists, form letters.

TABLE 1: *IBM Memory Expansion Device Driver Options*

MODE	NORMAL	ENHANCED	COMPATIBLE
DEVICE = [1]	PCJRMEM.COM	PCJRMEM.COM /E	PCJRMEM.COM /C
MEMORY (KB)			
User[2]	76 + exp	44 + exp	exp – 4
Video	16	32	96
VIDEO MODES			
All text	yes	yes	yes
160 by 200, 16 color	yes	yes	yes
320 by 200, 4 color	yes	yes	yes
640 by 200, 2 color	yes	yes	yes
320 by 200, 16 color	no	yes	yes
Graphic page(s)	1 @ 16KB	2 @ 16KB	6 @ 16KB
		1 @ 32KB	3 @ 32KB

[1] **Statements to place in the CONFIG.SYS file.**

[2] **The term "exp" means the amount of expansion memory installed. This amount is added to any memory in the main system unit that is available to user programs.**

The default creates a 16KB video area and defines the start of user memory at 52KB.

MEMTEST must be run before any other program to test all free memory in the computer. The program automatically skips over any memory occupied by DOS at the low end of the memory.

MENU is the control program that can be used to tie the other programs of Treasure Chest together for easy access. This master program allows single-key access to most other modules and controls the settings of several toggle functions.

PARTEST runs several tests on the IBM and Tecmar parallel ports.

QSORT uses Hoare's Quicksort algorithm to sort data entries within a file. The column to use, the order of sorting (forward, reverse), and input and output files are command options.

ROLODEX looks very much like its real-world counterpart. It is menu-driven and a snap to use. Its data files can be processed by other programs to provide other forms of output, such as form letters.

STOCK is an inventory program that tracks entries by stock number. Data may be modified and searched for from a main menu interface.

TTT plays a mean game of two- or three-dimensional Tic Tac Toe.

Anyone who is considering upgrading a PC*jr* to near-PC compatibility should become familiar with which video modes are needed for planned applications. It is advisable to use the smallest video buffer necessary so that the greatest amount of memory is available for applications.

On a price/performance basis, the Tecmar expansion units may edge IBM's because they provide some useful, additional functions and greater memory capacity. The *jr*Captain, in particular, can answer several immediate needs (clock, printer, and memory) in one convenient, reasonably priced unit.

The IBM options software offers greater flexibility in the selection of video buffers for a variety of needs. That benefit is balanced on Tecmar's side by its Treasure Chest software that is bundled with *jr*Captain. The IBM PC*jr* Memory Options software is bundled with the memory attachment and is not available separately.

IBM PCjr 128KB Memory Expansion Attachment: $325
Power Expansion Attachment: $150
IBM Corporation
5201 S. Congress
Boca Raton, FL 33432
CIRCLE 500 ON READER SERVICE CARD

jrCaptain, including Treasure Chest: $395
jrWave: $375 for 28KB; $495 for 256KB
Tecmar Inc.
Personal Computer Products Division
6225 Cochran Road
Solon, OH 44139
216/349-0600
CIRCLE 499 ON READER SERVICE CARD

Augie Hansen owns Omniware, a Denver-based software development company. He is the author of Chrome Ranger.

PC BRAND™'s CRAFTSMAN™ TOOLS FOR "C" SHOPS

C CROSS COMPILERS

Portability to 16-Bit Just Got A Lot Easier

PC BRAND now carries an assortment of cross compilers to move products from larger host machines to the PC-DOS or MS-DOS™ environment. Take a glance at the list below.

Cash in on products already developed on your bigger machines by rapid transit to the burgeoning world of PC owners.

Or take advantage of big machine services and utilities to do development work for the PC market, and only then download the results. A quick mental calculation will convince you that productivity gains will quickly trade off these costs. Each is custom-made to your specification. Call for availability.

MS-DOS or PC-DOS Target:	Product Code:	Price:
VAX/VMS Host	L10VM#	**$5000**
VAX/UNIX Host	L10UX#	**5000**
MC68000/UNIX Host	L1068#	**3000**
Altos 586 Host	L10AL#	**3000**
Hewlett Packard-UX	L10UX#	**3000**

Expand Your Sales to the CP/M World

They have vanished from the headlines, but there are over a million CP/M™ machines still humming across the land. Yet *InfoWorld* reports that "there's no new CP/M software". If you are part of the problem, here's a money-making way to become part of the solution. Convert your MS-DOS PC-DOS products with our CP/M-Z80 targeted cross compiler. For a few dollars, you'll double your market in a hurry.

CP/M-Z80™ Target:	Product Code:	Price:
MS-DOS or PC-DOS Host	L2Z80#	**$500**

C-SPRITE

Lattice's Own Debugger for Lattice® C

This versatile companion to your compiler gives you the best of both worlds for an out of this world price. Hand it a COM or EXE file produced by the Lattice Compiler (using the -d option) and C-Sprite™ will speak your language: your function names, your variable names, your data types, and the line numbers from your source code. At the same time, if you want to scrutinize just what machinations the compiler (or an assembler) has contrived, you can get a close-up view of machine addresses and machine coded instructions.

You can set simple breakpoints using symbols or addresses, or submit clusters of commands to be executed at the breakpoints, or set commands that execute until a condition is met.

You already know how to converse with C-Sprite, if you are familiar with Microsoft's Debug. Lattice began with that well-known command language, and then added to it considerably: You can work with data in hex, as you might expect, and you can also differentiate between C's data types, causing the debugger to treat addresses as pointers, or strings, or long integers, etc., both in display and entry. C-Sprite even has macros — use your source code variable names in a macro to dump the contents of entire C structures, for example. And you can debug through one of the COM ports with a second terminal so as not to disturb your program's display screen. What's more, if you link with Plink86™, C-Sprite can even tackle overlays.

Program doctors will find plenty of implements to rummage through in this kitbag.

Product Code: L2300# Price: **$175**

WE ARE EXPERTS IN EXPORT

PC BRAND ships all over the world. We'll prepare the export documents and ship to you or your agent at the port of entry you specify. Phone (212) 410-4000 or Telex 667962 (SOFT COMM NYK) with your order and preferred shipping method. We will compute the total charges for you to prepay by funds wired to Chemical Bank, 126 East 86th St., New York, N.Y. 10028, Account #: 034-016058, and your products will be shipped immediately.

TEXT TOOLBOX™ #1

These Utilities Work Wonders of Organization

Welcome to "grep", "wc", "ed" and "diff", tools you will reach for as routinely as "copy" once you come to know them. Unix™ boasts a number of muscular utilities that are migrating to the PC world. Lattice has assembled this cluster of the most useful text management tools into a single package.

"Grep" looks for text patterns in any number of files. Its powerful expression syntax goes far beyond your text editor's search command. Use of "(.+)" with "*.c" will find in all files with ".c" extensions all lines with parenthesized expressions, no matter how many characters lie between. Want to find all function calls? Look for all occurrences of, say, a global variable throughout a program system? Search for all programs in a directory, use paths to other directories? Find all files on a disk? "Grep" will grab them all.

"WC" counts lines, words, and characters in a file and has a checksum independent of machine character sets so you can test whether a file has een transferred successfullybetween computers.

"Ed" is similar to the well-known Unix editor. It offers search and replace with "Grep's" syntax, block move, read and write, optional line numbering, append, insert, delete, and this unusual facility: you can instruct "Ed" to apply a file of commands to any number of target files, even complicated changes and text additions, such as those created by "Diff".

"Diff"? You've probably tried to write one (and then discovered how tangled the logic gets). "Diff" compares text files line for line and reports differences. It uses complex algorithms to re-synchronize between files after disparities involving any number of lines are found. And it outputs is a precise list of instructions telling what to do to make two files exactly the same, a list which can be handed to "Ed" to do the job!

You'll ultimately find such assistance indispensible. Like having a librarian to sort out the confusion every day and keep your work tidy.

Product Code: L2220# Price: **$120**

CVUE

A Text Editor to Make Your Own

CVUE is a neat screen oriented text editor which does most of the things that a good editor should do, such as automatic scrolling vertically and horizontally, insertion and overtype entry modes, block delete, undelete and move, and full DOS 2.0 directory path name support in reading and writing files.

It is easy to learn with a comprehensive command menu screen which makes the documentation an ornament. It was written by the Lattice programmers who felt forgotten by the folks who write WP software. They needed easy entry of non-display characters such as control codes and escape sequences, not footnotes. Indenting and Undenting of block structures loomed larger than italic printing for them. Pattern searching won out over spell checking. So CVUE was born.

CVUE has its limitation. It only supports in-memory text files, but with memory at today's prices, creating and maintaining files of over 500 KBytes long is practical. Anyway, modular source code of structured programs never gets nearly that big. As compensation, CVUE is very compact and fast. It actually runs in computers with only 64 KBytes of memory and uses no tediously slow overlays to perform its full function repertoire.

The power of CVUE is its ease of customization. Even with only a binary license, full customization of the keyboard editing commands is offered. And when you take advantage of the Source Code option (found elsewhere in this ad.) the resultant editor can be made truly your own.

Product Code: L2240# Price: **$100**

CURSES

A Screen Management Interface to Swear By

Curses manages the screen of the IBM PC in the same fashion as the curses utility of Unix and similar operating systems. Use it to adapt programs which call Unix's curses functions for screen management, and need the equivalent library when moved to the PC for recompilation. Or use it when creating software on the PC to assure that it is Unix compatible.

Curses is a library of eighty-four functions and macros which can keep any number of screen images in memory.

Within a screen, Curses employs a vast function set to get characters, wrap lines, scroll, blank lines, highlight — virtually every tool needed to update the screen. The product supports color, and all four memory models. In keeping with the terminal orientation of Unix curses, the physical screen is re-painted (at high speed) only when your program calls a refresh function.

Writing screen management code leads to unspeakable snarls and expressions. Swear off! Let Curses clean up your language.

Product Code: L0850# Price: **$125**

dB-C

Switch from dBASE to C for Power and Speed

There are a lot of dBASE™ file users out there. Most just maintain data bases and use dBASE's limited reporting facilities. They're not programmers, so they don't use the dBASE programming language. But they'd like a greater return on their efforts, and that's a business opportunity that makes this product important to you.

dB-C™ links C to dBASE. It creates files which exactly replicate dBASE file design. So dBASE can read and update them. And the reverse. dB-C can use any files created by dBASE. Now C and dBase can operate on the same data bases interchangeably.

This opens up the widespread culture of dBASE installations to exploitation. As a C programmer, you can replace the resident dBASE language with speed! Both C's and your own: you no longer have to write every line of code, because now you've unlocked C's vast storehouse of off-the-shelf libraries and utilities.

And you add power: dB-C lets you open and process eight index and data files at once.

Use dB-C for custom work. Or design generalized programs for manipulation and reporting of dBase data bases. Or use it on its own. It's a complete file manager for use with the Lattice C Compiler whether or not dBASE will ever be used in tandem. dB-C comes as a library of object modules, with versions for all four memory models. It's a complete set of ISAM routines (that parallel dBASE commands) which create and maintain index and data files. The manual discusses each in detail, and demonstration source files on your disk show how every function is used.

Two Versions:	Product Code:	Price:
Dbase II Compatibility:	L00II#	**$250**
Dbase III Compatibility:	L0III#	**250**

PRECISION SOFTWARE FOR PROGRAMMERS

SUPPORT DIRECT FROM LATTICE

Products with codes beginning "L" are all fresh from Lattice Inc., the premier software developer serving the C professional marketplace. PC BRAND has made special arrangements for Lattice to provide telephone support for these products directly.

Take advantage of this unusual opportunity. Buy from PC BRAND, and you get to talk to the experts themselves!

30-DAY MONEY-BACK GUARANTEE[a]

It gets better. We will return the purchase price[b] of any "L" or "N" coded product offered in this space if you decide it fails to meet your needs. You read that correctly. We want you to be completely satisfied with your purchase. Go through the manuals, *try out the products themselves*, and make a thoroughly informed decision whether they are right for you.

There are rules: We can make this offer in the U.S. and Canada only. For you to be eligible for refund or credit, we must *receive* your returned product within 30 days of its shipment. Everything but the packing must in our judgment be in 100% resaleable condition.

(a) Subject to terms and conditions of sale. (b) Not shipping and handling.

CODE SIFTER
Find the Hot Spots in Your Program

Fast is never fast enough; only miracles need apply. That's what's expected in today's marketplace, so our Code Sifter is an important tool to add to your workbench.

Code Sifter finds the trouble spots in your program. On its own, it will divide a COM or EXE file of any size into thirty-two equal partitions. Alternatively, you can specify the partition boundaries with addresses, or with symbols if your linker has produced a symbol map. Then tell Sifter to run your program. It samples your object file at precisely timed intervals and counts how many times it finds the instruction pointer in each partition. Job done, it reports the number and percentage of hits in each partition.

You are in for some surprises when you discover just how unbalanced the activity is likely to be, and that's why Code Sifter is so valuable. It profiles just where you can best spend time optimizing your code, or even converting to assembler subroutines.

Code Sifter has a number of monitoring options. You can tell it to include any combination of your program, DOS, and BIOS in its analysis. You can specify the sampling rate. Most important, you can tell Code Sifter the number of times to run a program, and between each run discard the less active ranges and repartition the hot spots, so that you zoom in ever tighter on small areas of code. Right down to the last byte! Literally. Try it on the sample program that comes with your disk.

Code Sifter. It will give you the racer's edge.

Product Code: N3100# — Price: **$119**

LMK
A Unix-like "Make" Makes Light of System Building

If you have ever built a complex system, you know the time loss and tedium of recompiling, rebuilding libraries and relinking modules because a snippet or two of code has changed. Batch files are no answer. You need batches of them to avoid redoing everything indiscriminately.

Instead, imagine making a change deep in a system, and simply telling Lattice's LMK™ to take over. No further thinking or keystrokes. LMK will rebuild your final product, however involved and complex, by doing just what is needed and no more.

How? You write a command file which expresses bottom to top all the elements comprising your system and all its dependencies: what gets compiled to make what object file using what options; what is built into libraries; what is linked into the final EXE file. Through the life of your system, LMK keeps track of the last time every action was performed. Run LMK and, tracking each branch, it looks only for elements which changed later than a dependent element further along the branch, using date and time information found in the file directory. Any source file newer than its object file, for example. Only those elements and their dependents are remade. All other instructions are bypassed.

The command file uses a simple, readable syntax — "prog.obj: prog.c $(HDRVS)", for example, says what source file this object file depends on, and fills the previously defined macro HDRVS into the expression, which here might be a list of files with hardware drivers, or in another case your preferred string of compiler options.

LMK does not care what programming language you use; it's not just for C. For that matter, LMK can apply to more than programming. It can be used for *any* set of tasks which can be accomplished through commands issued to the operating system. Try it for repeated re-assembly of lengthy documentation, or for *selective* re-consolidation of spreadsheets so that only dependents of changed supporting schedules get recalculated.

Wherever your imagination leads you, LMK will find the shortest path to get the job done. Minimum time, minimum effort software.

Product Code: L2100# — Price: **$195**

USED COMPILERS WELCOME
Trade In for the Latest Model Lattice C

Has your compiler run out of gas? Has your model been discontinued? Is it falling behind for lack of new parts?

Even if yours is in good shape, you have surely noticed there are more options and accessories produced to run with the Lattice C Compiler than any other. Don't do without these additives any longer. It's time for new license plates. Trade in your original disks and manual of any of the compilers below and we'll send you Lattice's most up-to-date model, the full bore 2.14 (at press time).

From then on you will be adopted by Lattice for full, direct support by their technical specialists.

	Price:
Product Code LU150:	
Microsoft MS-DOS/PC-DOS C	**$150**
Product Code LU200:	
Computer Innovations C86	**$200**
Manx Aztec C86	**200**
Mark Williams C	**200**
Wizard C	**200**
Digital Research C	**200**
Whitesmith's C	**200**

SOURCE CODE FOR SALE
Designer Originals For Your Special Needs

"One size fits all" turns away disappointed customers. When you cannot shoehorn your application into out-of-the-box software, we have the solution. For each of these products only, PC Brand now licenses source code, provided you buy (or have already bought) from us the object code counterpart. Take it in, let it out, and make it fit snugly to your needs. And, if you are new to C, you will learn a lot apprenticing this fully documented code from top designers.

	Product Code:	Price:
Lattice C Compiler Library		
for MS-DOS/PC-DOS	L9100#	**$500**
for Z80	L910Z#	**500**
C-Food Smorgasbord Library	L9200#	**500**
Curses Library	L9850#	**125**
dB-C Library		
for dBase II	L90II#	**250**
for dBase III	L9III#	**250**
CVUE Screen Editor	L9240#	**350**
Text Toolbox #1	L9220#	**120**

TERMS AND CONDITIONS OF SALE

Licenses: Each price is for a license to use a product on a single computer and does not constitute product ownership. Ask about availability of other terms for multiple machine use at a single site. Products coded "#" may be used to create programs for distribution without royalty payments or additional licenses, provided said programs do not substantially replicate the products themselves.

Compatibility: PC BRAND's standard products are designed to operate with the IBM PC, PC XT or AT under PC-DOS and require no more than 128K of RAM unless indicated.

Return of Non-'L'-Coded Products: Acceptance for refund or credit only by prior authorization subject to a re-stocking charge. Software diskettes are delivered in sealed envelopes and are unreturnable if seal is broken. Defective products will be replaced in whole or part.

Payment: We honor MasterCard & Visa (no surcharge), wired funds, checks in advance, and COD for cash or certified check ($5 fee). Purchase orders accepted from corporations and institutions at our discretion; 2% per month added to balances unpaid after 30 days.

Shipping & Handling: U.S.: UPS Surface: 1st product $6, each add'l $3.00. UPS 2nd Day Air: 1st product $10, each add'l $4.50. UPS Next Day Air or Federal Express: 1st product $18, each add'l $6. International: Charges dependent on destination and method of shipment.

We're as Near as Your Telephone. Call us at...

800-PC BRAND T2

That's (800) 722-7263. In NY State call (212) 410-4001.

PC Brand, 345 E. 86th St., New York, N.Y. 10028

Prices, terms, and specifications subject to change without notice.

CIRCLE NO. 170 ON READER SERVICE CARD

BACKUP. AND WHY PEOPLE DON'T.

An Introduction To InfoTools' BAKUP™: The First Intelligent, High-Speed Software Backup For Hard Disk Systems.

Backing up important business files from hard disk is a pain—because traditional backup solutions are slow, unselective and, too often, too expensive. That's why people too frequently just cross their fingers and hope for the best . . . and invite the worst.

The BAKUP Solution: First, It's Fast.

BAKUP, the first intelligent backup, restore, and cataloging software system, can save you from disaster by securing your critical data—fast. With BAKUP, the time-consuming copying of hard disk files to floppies is slashed by up to 2½ times over other alternatives. The first backup of a 10-megabyte disk takes less than half an hour—and after that, daily backup normally takes only minutes.

Second, It's Smart.

InfoTool's designers have backed up speed with intelligence—in an exceptionally easy-to-use, menu-driven system.

- Simply select what you want backed up, and BAKUP does it without further instructions—and BAKUP will tell you beforehand how long the process will take and how many diskettes you'll need.
- While securing your files, BAKUP creates a unique catalog that keeps track of where your files are—so when it comes to restoring, you simply browse through the catalog and mark the desired files; then BAKUP tells you which diskettes to insert.

The BAKUP Solution: $149.95 In Software, Not Thousands In Hardware.

InfoTools' BAKUP software system, for its user practicality, its management reporting capability, and its unusually low price, is a revolutionary alternative to streaming-tape hardware that can cost 10 to 15 times more. If you already have another file-oriented backup device such as a high-capacity floppy, nonstreaming tape, or removable cartridge drive, BAKUP will be a sensible addition for more speed and convenience.

Put An End To Risk-Taking With Our 30-Day, Money-Back Guarantee.

BAKUP is the right choice for IBM PC/XT, AT, or compatible systems with mass-storage management needs. And it's as easy and risk-free to get as it is to use. Just fill out the coupon below or call our toll-free order number. Use your VISA or MasterCard. If, within thirty days, you find it's not right for you, for any reason, we'll give you your money back. It's that simple. Dealers, OEMs, or corporate buyers should call (408) 725-1151 to inquire about our volume discounts.

So, uncross those fingers. Pick up a pen or a phone and order BAKUP today.

Okay, I Promise To Start Backing Up My Hard Disk Files.

Send me ____ copies of BAKUP, complete with documentation and diskette labels, at $149.95 each, backed by a 30-day money-back guarantee. Add $5.00 shipping and handling. In California, add sales tax.

Name _____
Title _____
Company _____
Address _____
City _____ State _____ Zip _____
Phone _____
VISA/MasterCard # _____
Expiration Date _____
Signature _____
☐ Check or money order enclosed.

INFOTOOLS
Steps Ahead In System Backup And Restore.

InfoTools
10044 S. DeAnza Blvd.
Cupertino, CA 95014
(408) 725-1151

Toll-free telephone for orders only,
6 AM to midnight (Pacific time)
In California call 1-800-672-3470 Ext. 801
Outside California 1-800-538-8157 Ext. 801

CIRCLE NO. 144 ON READER SERVICE CARD

Dipping into Directories

Once a subdirectory is opened, DOS will treat it as just another file.

TED MIRECKI

One of the major functions of an operating system is to keep track of the disk locations of information. When it's time to resume work on his immortal prose, an author would prefer to tell the word processor that his text is in file WHATSUP.DOC, rather than on track 32, head 1, sector 4.

Most of the time, computer users are content to let the operating system keep track of the disk locations where it squirreled away each of their files. Sometimes, however, systems programmers need to access directly the information kept in the disk's directory and File Allocation Table (FAT) to perform a variety of services not provided by the DOS directory management functions. In ancient times (that is, before DOS 2.0), a program always knew where to look, because the information that related file names to disk addresses was kept in a single directory that began and ended in the same location on every disk. Simplicity has its price, however: a fixed-length directory could hold only a fixed number of files. With many small files, it was possible for the directory to fill up before the disk's data space was used up.

In DOS 2.0, the limitation on the number of files per disk is removed by allowing directory information to have variable length and arbitrary location on the disk. So that DOS knows where to start a directory search, each disk has a root directory in a fixed location and of a fixed length. Entries in the root directory may point to files or to subdirectories, which, in turn, point to other files and/or subdirectories, and so on. These subdirectories may be of any length and may reside anywhere on the disk, because they are organized in the same way as regular data files. The DOS documentation claims that subdirectories can be read as files, and since DOS 2.0 introduced some convenient "extended" file management functions, it should be easy for programs to access directories, right? Well, not quite.

To be sure, subdirectories can be processed as files, but the process turns out to be more involved than one would anticipate. Implementing useful directory functions takes a fair degree of proficiency with the Macro Assembler and familiarity with the back pages of the DOS manual that describe the disk allocation scheme, the directory format, and the INT 21H functions.

PROBLEM-SOLVING

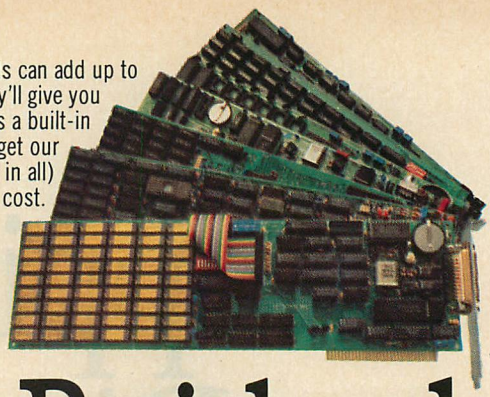
Before getting into the specifics of a program for reading subdirectories, let's first take a look at some of the general problems involved. The first problem is that the entry for a subdirectory in its parent directory is marked with attribute 10H, making it a hidden file to most DOS functions. That is why a subdirectory cannot be deleted, copied, typed, or renamed. Nor can it be opened with the DOS 2.0 function 3DH (open file, described in appendix D of the DOS 2.0 manual): trying to do so results in error code 5, access denied. The manual says that this function will open any normal or hidden file, but does not mention a subdirectory.

Appendix D further describes function 43H, which changes a file's attribute. Perhaps the subdirectory's attribute could be temporarily changed to zero, processed as a normal file, then changed back? No such luck: function 43H has no effect on the subdirectory bit of the attribute. Attempts to turn it off are ignored, and attempts to turn it on, even on a file that already is a directory, produce error code 5 again.

It is possible to turn on any of the other attribute bits; so, for example, a user can mark a subdirectory as hidden, system, read-only, or any combination, but it remains a subdirectory and inaccessible to the extended file functions of DOS 2.0. Incidentally, a subdirectory that is marked system and/or hidden will not appear in the listing of its parent directory, but otherwise acts as a normal directory: it can be made the current directory, files can be added or deleted, and it can be removed when empty. Turning on the read-only bit of a directory's attribute has no effect.

So, although subdirectories are a feature of DOS 2.0, they cannot be processed with the file-handling functions. Opening a subdirectory file with a function that does not use a File Control Block (FCB) would not work anyway, because of the next problem. Strangely enough, the "traditional" file functions carried over from earlier DOS versions work just fine. A subdirectory may be opened with function 0FH, using an extended FCB with an attribute byte of 10H (see appendix E of the DOS 2.0 manual). First problem solved.

Our Captain™ or other multifunction boards can add up to 384K bytes of memory to your IBM PC. They'll give you additional serial and parallel ports as well as a built-in clock/calendar. With each board, you'll get our Treasure Chest™ of Software (24 programs in all) at no extra cost.



Tecmar's Lab Master™ is ideal for process control and the tracking of laboratory experiments in industrial and scientific applications. It converts analog data to digital data, and back again. Sophisticated timing is accomplished throughout this process by Lab Master's timer/counter circuitry (AMD 9513). The Lab Master has 24 digital I/O lines. 12-bit resolution is standard while 14- or 16-bit resolution is optional. Additional A/D and D/A channels are also available.

For your graphics needs, Tecmar offers the Graphics Master.™ With it, your PC gets graphics support for most popular software packages like Lotus 1-2-3,™ Open Access,™ Auto CAD,™ and all IBM color software. Graphics Master works in high resolution color (640 x 400) or Monochrome (720 x 700).

Peripheral Vision

When your computer needs peripherals, it pays to go with the company that has the peripherals and the vision. Tecmar.

In all, Tecmar has more than 150 peripherals and add-ons for your computer. So, when you're ready to expand, ask for Tecmar. Or write for our brochure.

Tecmar, Inc., 6225 Cochran Road, Solon, OH 44139 USA, Tel. (216) 349-0600, Telex 466692.

Tecmar International, Inc., Avenue de la Tanche, 2, B1160 Brussels, Belgium, Tel. (02) 660-44-51, Telex 25387.

Tecmar

THE POWER BEHIND THE PC

For large memory storage, Tecmar offers your PC a variety of Hard Disk Drives with capacities up to 74 megabytes. These include Removable Disk Drives for unlimited hard disk storage.

And Tecmar also makes the MacDrive,™ Removable or Fixed Hard Disk Drive for the Macintosh.

The second problem is that a subdirectory's length is recorded as zero in its parent directory. When the subdirectory is opened as a file, the zero size is copied into the FCB, and all read attempts fail because DOS assumes that the user is trying to read past the end of file. The length of the directory must be determined, and that value must be set into the FCB after opening the file and before the first read. DOS must know how long the directory is when it attempts to perform directory searches, and CHKDSK reports the total size of all subdirectories on a disk, so it too must be able to determine the size of each one. A program can also determine the size of a directory, by tracing the cluster allocation chain in the File Allocation Table (FAT), as described in appendix C of the DOS 2.0 manual. Since a subdirectory always occupies an integral number of clusters, counting the clusters in its allocation chain is a sure way of determining its size, provided that the length of each cluster is known.

Because the FAT is not a file, its location and length are not recorded in any directory. Therefore, none of the DOS file-handling functions can be used to read the FAT into memory; instead, the user must know its absolute disk address and read it in with INT 25H (absolute disk read).

DOS requires that the FAT begin immediately following the boot record on each disk, and that the boot record begin on logical sector zero. On all current floppy and hard disks, the boot record is one sector long, so the FAT begins on logical sector 1. To allow for different disk allocation schemes in future versions, DOS 2.0 and above write the characteristics of the disk in a BIOS Parameter Block (BPB) near the beginning of the boot record when the disk is formatted. The contents and format of the BPB are described in chapter 14 of the DOS 2.0 manual. From the BPB, the user can get most of the information needed to find the FAT and begin tracing its cluster chain, but there is one catch: a disk formatted with DOS 1.0 or 1.1 does not have a BPB in its boot record. Although the earlier versions did not support subdirectories, there is nothing to prevent DOS 2.0 from being used to create a subdirectory on a disk formatted by an earlier version.

Fortunately, there is a way of obtaining the FAT information for any disk. Any one of the functions 1BH, 1CH or 36H (get disk allocation information) will return the total number of clusters mapped in the FAT, the bytes per sector, and the sectors per cluster. The logical length of the FAT in bytes is the total clusters times 1.5, since the entry for each cluster is 12 bits, or 1½ bytes. The physical length of the FAT in sectors is the byte length divided by bytes per sector, rounded up to the next integer; the length of the cluster is the number of bytes per sector times the number of sectors per cluster.

Another value that is needed for the FAT scan is the starting cluster number of the subdirectory file. This is obtained by functions 11H or 12H (get first or next directory entry), either of which return the directory entry of a file in its parent directory.

The procedure for reading a subdirectory may be outlined as follows:

1. Change the current directory, if necessary, to the parent of the directory to be processed.
2. Determine the disk characteristics with function call 1CH, and read the FAT with INT 25H.
3. Using function 11H, get the directory entry for the desired subdirectory. Save the starting cluster number.
4. Open the subdirectory as a file with function 0FH, using an extended FCB with 10H attribute byte.
5. Trace the file's cluster chain through the FAT as described in appendix C of the DOS 2.0 manual. For each cluster in the chain, add the cluster length in bytes to the file length. At the end of the chain, insert the calculated length into offset 10H from the start of the normal FCB (17H from the start of the extended FCB).
6. Set the FCB record length and record number fields to the appropriate values, read the data from the subdirectory file, process it, and write it out as needed.
7. At the end of the process, close the file if any writing was done to it.
8. If more subdirectories are to be processed, go back to step 3; otherwise, terminate the program.

Note that in the above sequence, the FAT must be kept in memory throughout the process. If memory space is at a premium, the FAT could be read into the same buffer as the contents of the directory, because the FAT is not necessary after the cluster chain is traced. But in that case, if more than one directory is being processed, the FAT contents must be recovered by looping to step 2 from step 8.

UNLIMITED POSSIBILITIES

What can be done with subdirectory files and why would you want to do anything with them? First of all, be aware that what can be done is virtually unlimited. Once the subdirectory is opened, DOS treats it as just another file and will cheerfully perform whatever file operations are asked for, without regard for the havoc it may wreak on the integrity of the files recorded in that directory. The user may overwrite the directory with a letter to Aunt Tillie, Uncle Ed, and the cousins; extend it with any information desired; truncate it; or delete it entirely.

The question, therefore, is not what can be done, but what should not be done. After all, the user is playing on the operating system's turf, and if he expects it to pick up the pieces after he is done, he had better play by its rules. Those rules state that a directory contains 32-byte entries, in a very specific format, giving information about files and pointers to their locations on disk. If the directory is to continue to function as a directory, it had better be left in the proper state.

Within these limitations, plenty of useful processing still can be done. One useful function is to look for deleted files and attempt to recover them. A deleted file is identified by 0E5H in the first byte of a directory entry; the remainder of the entry still contains the rest of the name and all of the other directory information. The first cluster (only) of the deleted file can be located on disk, provided it has not been allocated to some other file since the deletion. The user should note, however, that the chain of clusters after the first cannot be traced, because those clusters have been deallocated and are now part of the disk's free space.

Another possibility is to order the entries in the directory in some particular sequence, say alphabetically by file name, so that the directory list is displayed in a convenient order without the need to sort it at each listing. Perhaps a more useful order would be to place the most frequently accessed files, whether they are programs or data, at the top of the directory list, so the system can find them more quickly.

DOS does not care about the order of files in the directory, with one exception: in a subdirectory, the first two entries must be subdirectories with names of one and two periods respectively. The first points to the disk location of the beginning of this subdirectory, the second to the beginning of its parent directory (the pointer is zero if the parent directory is the root).

Several subdirectories may be opened at once, just as several data files may be processed simultaneously, as long as each is in the current directory of its disk drive. Remember that subdirectories must be accessed through an FCB, not with path names, and an FCB

CODING .COM FILES WITH MULTIPLE SEGMENTS

In a program loaded from a .COM file, all references to code and data must be addressable as offsets from one base: the start of the Program Segment Prefix (PSP). To ensure such addressability, the code and data are usually defined in the same segment. Since the assembler will operate more efficiently when data labels are defined before they are referenced, the usual practice is to define the data ahead of the code and to begin the program with an instruction to jump around the data definitions in order to get to the code section. This structure often is encountered when some of the DOS external command files or some commercial programs are unassembled.

Two problems with this are (1) the programmer must know how much data space to reserve when writing the program and (2) the data definitions take up as much room on the disk as they do in memory after the file is loaded. So if the program consists of 10KB of data space followed by 1KB of code, the resulting .COM file takes up 11KB on disk. If the data area does not need to be initialized, the programmer could save the 10KB of disk space by defining the data area after the code and declaring a label identifying its beginning. Of course, then the inefficiencies of forward referencing would become apparent.

There is a way to get the best of both worlds. Figure 1 shows the general structure of a program that is meant to be converted to a .COM file, but which consists of separately named data and code segments. The data segment is defined first, but the linker will place the data segment after the code segment in the load file, *provided that the code segment name alphabetically precedes the data segment name.* The alphabetic ordering of segments in load files is not expressly documented in IBM's or Microsoft's manuals for the DOS Linker.

Remember that when the OFFSET operator on a label from the data segment is used, it must be prefixed with the group name. Failure to do so is not flagged as an error by the assembler, but the resulting value is the offset from the start of the segment, not the start of the group, and will be in error by the length of the code segment. This can be a difficult error to catch.

Note that other references to the data labels, including loading their offsets with LEA instructions, do not need to be prefixed with the group name. Surprisingly, references to labels in the PSP need not be prefixed with either a group name or segment override. Even though such labels are in the code segment, not the data segment, the assembler seems to be smart enough to know that they are accessible as offsets from the DS register.

In a COM program, a stack segment is not defined, nor is SP reset, because the program loader assigns stack space at the high end of the segment in which the program is loaded. The bottom of the stack is at offset FFFEH, and it grows toward low memory as far as needed. If an open-ended data buffer follows the code, writing to that buffer must be limited to an address below the lowest expected stack top. *—TM*

FIGURE 1: *General Structure of Multisegment .COM Program*

```
Keywords are in caps, user-supplied names in lower case


groupname GROUP     codname, dataname
          ASSUME    CS:groupname, DS:groupname
          ASSUME    DS:groupname, ES:groupname

dataname  SEGMENT   ;specify alignment, class if desired
value1    DB        20          ;define data as necessary
value2    DW        1024
  .
  .
buffer    LABEL     WORD        ;undetermined number of words
dataname  ENDS

codname   SEGMENT
          ORG  5CH              ;define labels in PSP
param1    DB   12 DUP(?)        ;1st command line filename
          ORG  6CH
param2    DB   12 DUP(?)        ;2nd command line file name
          ORG  80H
parmlen   DB   ?                ;length of command line
parmline  DB   127 DUP(?)

          ORG  100H             ;code origin
procname  PROC
            .
            .
          MOV  BX,OFFSET groupname:buffer    ;need prefix
          LEA  SI,value2                     ;no prefix
          MOV  [BX+SI],ax           ;can address beyond end
            .                       ;of defined data
            .
          CMP  parmlen,value1                ;no prefixes
            .
            .
procname  ENDP
codname   ENDS
          END  procname             ;specify entry point
```

This program, with separately named data and code segments, is meant to be converted to a .COM file.

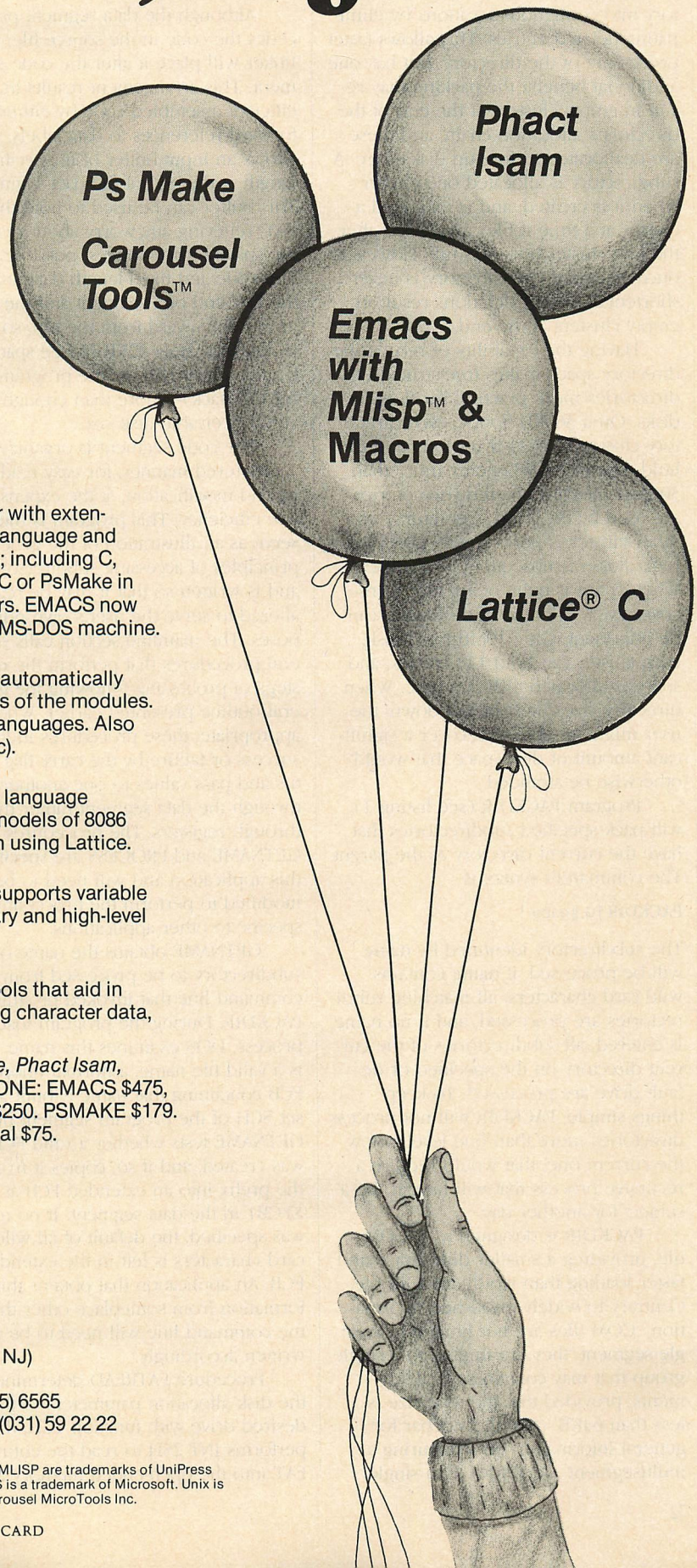
can only process files in the current directory. But by performing CHDIR functions (AH=3BH) at appropriate times, entries could be read from one directory and written to a second; this would effectively move files or whole sub-trees from parent to parent without actually copying their contents.

A file's entry can be left in more than one directory. Programs can then access it from any directory that lists it without the need for path names. This will not cause any problems as long as the length and starting cluster of the file are not changed. Reading and writing in place are acceptable, although writing will update the file's time and date in only one directory. Also, CHKDSK will complain about files that are cross-linked, and will report the wrong file count but the correct space utilization.

Creating a subdirectory with a program is possible but tedious, because the user must insert the first two entries that point to the directory just created and to its parent. The location of a newly created file cannot be determined until that file is closed, so the user would have to create a file, write something into it so the length is not zero, close it, look up its starting cluster and that of its parent in the parent directory, and then open it again to insert the two special entries. It would be a much simpler matter to create the subdirectory with DOS function 39H, which handles all of these details easily, than to open and process the subdirectory as an existing file.

Hackers, rejoice!

Finally, a programmable multi-window editor for MS-DOS.™



EMACS

UniPress EMACS™
Famed Gosling version. Multi-window, text editor with extensibility through the built-in MLISP programming language and macros. Dozens of source code MLISP functions; including C, Pascal and MLISP syntax checking. Run Lattice C or PsMake in the background and Emacs will point to any errors. EMACS now runs on TI-PC, IBM-PC AT, DEC RAINBOW or any MS-DOS machine.

PsMake
UNIX™ style 'make' utility, includes facilities for automatically *rebuilding* programs based on inter-dependencies of the modules. Includes rule scripts for many popular MS-DOS languages. Also includes many UNIX style tools (ls, cat, touch, etc).

Lattice C
'The' C compiler for the serious developer. Full C language producing the best code for the 8086 family. All models of 8086 supported. Emacs, PsMake, and Phact all written using Lattice.

Phact Isam
Multi-key ISAM for MS-DOS. Uses b + tree, and supports variable length records. Includes full Lattice linkable library and high-level functions.

Carousel Tools — UNIX-like facilities
Set of over 50 programs, utilities and software tools that aid in general file handling, searching and manipulating character data, and writing programs and documents.

FULL SYSTEM *[includes Emacs (object), PsMake, Phact Isam, Lattice C & Carousel Tools]* — $1,299. STANDALONE: EMACS $475, source — call for terms. Lattice C $425. PHACT $250. PSMAKE $179. CAROUSEL TOOLS $149. One month EMACS trial $75.

UniPress is a major publisher of Unix Software.

Call or write for more information.

UniPress Software, Inc.

2025 Lincoln Highway, Edison, NJ 08817
201-985-8000 • Order Desk: 800-222-0550 (outside NJ)
Telex: 709418 • Mastercard and Visa
Japanese Distributor: SofTec, Telephone: 0480 (85) 6565
European Distributor: Modulator SA, Telephone: (031) 59 22 22

A PROGRAM TO READ SUBDIRECTORIES

The program presented below performs a more modest, but no less useful, function. It packs the entries in a subdirectory into contiguous locations by eliminating deleted entries. This allows faster processing of the directory, but has one additional benefit: the packing may result in empty clusters at the end of the directory's allocation chain, and these are deallocated to free up disk space. A subdirectory is allocated one cluster when it is created, and is extended a cluster at a time if files are added after the first cluster becomes full. However, once extended, the directory is never shortened, even if deletions result in empty clusters at the end.

Having the capability of recovering directory space makes tree-structured directories more practical for floppy disks. On a 360KB floppy, each directory cluster is 1,024 bytes long and can hold 32 entries. Whenever more than 30 user files are entered (two entries are used by the . and .. pointers), another cluster is permanently allocated, regardless of subsequent deletions. When space is tight, being able to recover every kilobyte of disk space can be important. On a 10MB fixed disk, each cluster can hold 128 entries, and space is allocated 4KB at a time. When directory sizes must be cut down, the user might be able to recover a significant amount of disk space that would otherwise be stranded.

Program PACKDIR (see listing 1) will pack specified subdirectories that have the current directory as the parent. The command's syntax is

PACKDIR [d:][name]

The subdirectory identified by **name** will be processed; if **name** contains wild-card characters, all matching subdirectories are processed, and if no name is entered, all subdirectories of the current directory on the specified or default drive are processed. To keep things simple, PACKDIR will not process directories more than one level below the current one; that would require a recursive process that will have to be a subject for another day.

PACKDIR is designed as a .COM file, providing a smaller disk image and faster loading than would an .EXE file. Contrary to widely published information, .COM files are not limited to a single segment: they are limited to a single group that may contain several segments, provided that the total size is less than 64KB. (See the sidebar for general information on structuring a multisegment .COM file). In a single segment containing both code and data, there is usually a compromise between efficient space utilization and efficient code, whereas separate code and data segments can optimize both of these.

Although the data segment precedes the code in the source file, the linker will place it after the code segment. This arrangement results in more efficient assembled code by eliminating forward references to data labels, yet allows an input buffer of unspecified length at the end of the data segment. This buffer can be used to hold the FAT, relieving any worry about reserving space for the longest possible FAT. The code and fixed-length data are in the low end of a segment and the stack grows downward from the top, so there is approximately 62KB of free space in between (say 1KB for the program, 1KB for the stack)—more than enough for any conceivable disk size.

The code segment is organized in a structured manner, for easy readability and modification, at the expense of raw efficiency. This program is meant to serve as an illustration of the general principles of accessing subdirectories, and is written so that it may be readily altered to serve the user's own purposes. The mainline section calls several procedures that perform the major steps of processing following the general outline presented earlier. Where appropriate, these procedures indicate success or failure by the carry flag status and pass values to one another through the data segment rather than through registers. The procedures GETNAME and PROCESS are specific to this application and will need to be modified to perform the processing specific to other applications.

GETNAME obtains the name of the subdirectory to be processed from the command line that invoked program PACKDIR. During the program loading process, DOS examines this name. If it is a valid file name, a normal unopened FCB containing that name is built at offset 5CH of the program segment prefix. GETNAME tests whether a valid FCB was created, and if so, copies it from the prefix into an extended FCB at label XFCB1 in the data segment. If no name was specified, the default of all wild-card characters is left in the extended FCB. An application that obtains this information from someplace other than the command line will need to be rewritten accordingly.

Procedure FATREAD determines the disk allocation parameters of the desired drive with function 1CH, then performs INT 25H to read the entire FAT into the open-ended memory area at the end of the data segment. Here it is assumed that the FAT begins on logical sector one. This is true for all current floppy- and fixed-disk formats, but the procedure could readily be extended to the general case by obtaining the FAT's starting sector as recorded in the BPB in the partition boot record (logical sector zero) of the drive.

OPEN is the procedure that first performs function 1AH to set the Disk Transfer Address to the extended FCB at label XFCB2, then calls function 11H to search the current directory on the specified drive for a name matching the one in the FCB at label XFCB1. If found, DOS builds an extended FCB at the DTA. Using the directory entry structure overlaid on this address, the procedure checks to see if the name identifies a subdirectory, because both normal files and subdirectories are found when searching with attribute 10H.

If the matching name is not a directory, or is one of the special pointer entries, the procedure continues searching with function 12H until it finds the next matching name that is a user subdirectory. When one is found, the pointer to the directory's first cluster is saved, and the directory is opened as a data file with function 0FH, using the FCB built by DOS at label XFCB2. In this open FCB, the file size field contains a value of zero, and must be set to the actual length before the data can be read. The name of the subdirectory is displayed by repetitive calls to function 2, rather than by a single call to function 9, because the name may contain the $ character (the string terminator for function 9). Upon return, the carry flag indicates whether a matching name was found: the carry flag is off if one was found, on if one was not found.

Procedure DIRSIZE determines the length of the subdirectory file in bytes and clusters by tracing the allocation chain through the FAT, essentially as described in appendix C of the DOS manual. It then inserts the size value into the proper double word of the open FCB. The size will overflow a single word only if the directory size exceeds 64KB, or 2,048 entries—not very probable, but safe nevertheless.

Whatever processing is to be done on the subdirectory entries is performed by PROCESS. The record size per read must be inserted into the FCB; here the length is set at 32 bytes, because the user is interested in only one entry at a time. In this program, each entry is read from the directory using the random read function, 21H, and it is written out to the same record number if it is not deleted. When an entry be-

ginning with zero is written out, the processing ends, because there are no more entries in this directory.

Finally, procedure CLOSE truncates the file to its new length and closes it. In DOS, closing a file does not change its length, no matter what the value of the file length in the FCB at close time. Changing the file length is done with function 28H, random block write, with a record count of zero and the new file length specified in the random record field of the FCB. Note that this length need not be a multiple of the cluster size; DOS will allocate an integral number of clusters anyway, and for directories, the file length recorded in the parent directory entry is immaterial. The new length in clusters is calculated only for purposes of displaying the old and new lengths of the directory.

As a bonus, PACKDIR includes the I2ASC routine that converts a 16-bit signed binary integer into ASCII digits. It is coded so that, with proper changes to the segment name, it may be included in any other assembly program.

Processing subdirectories is not too different from processing any other data file. The procedures suggested here by no means exhaust the possibilities, but are meant only as a catalyst. Remember, however, that there is one crucial difference between normal files and directories: if an errant program trashes a data file, most likely only that one file will be lost, but when playing with directories, one mistake can ruin an entire disk. It is best to experiment on throwaway copies of floppy disks (better yet, on a RAM disk) before trying to modify directories on good disks, or worse, on 10MB of a fixed disk.

Ted Mirecki has a master's degree in computer science and 20 years of experience in information processing. He is currently employed as a corporate planner.

LISTING 1: PACKDIR.ASM

```
TITLE       Program PACKDIR

            COMMENT "                          by Ted Mirecki

            Reads a sub-directory as a file, packs entries into
            adjacent locations, eliminating deleted entries, and
            deallocates empty sub-directory clusters.
"

COM         GROUP   CODE,DATA
            ASSUME  CS:COM, DS:COM, ES:COM, SS:COM

;*****************************************************************
;
;   STRUCTURES
;
;*****************************************************************

; Layout of a directory entry.

DIRENT      STRUC
DNAME       DB      8 DUP(?)        ;NAME OF FILE
DEXT        DB      ?,?,?           ;EXTENSION
ATTR        DB      ?               ;ATTRIBUTE BYTE
            DB      10 DUP(?)       ;RESERVED FOR DOS
DTIME       DW      ?               ;FILE UPDATE TIME
DDATE       DW      ?               ;FILE UPDATE DATE
STARTCL     DW      ?               ;STARTING CLUSTER
DSIZE       DD      ?               ;FILE SIZE, DOUBLE WORD
DIRENT      ENDS


; Layout of a normal FCB

NORMFCB     STRUC
DRIVE       DB      ?               ;DRIVE ID
FNAME       DB      8 DUP(?)        ;FILE NAME
FEXT        DB      ?,?,?           ;FILENAME EXTENSION
CURBLOK     DW      ?               ;CURRENT BLOCK
RECSIZE     DW      ?               ;RECORD SIZE FOR EACH READ
FSIZELO     DW      ?               ;LO-ORDER WORD OF FILE SIZE
FSIZEHI     DW      0               ;HI ORDER WORD OF FILE SIZE
FDATE       DW      ?               ;DATE OF LAST UPDATE
            DB      10 DUP(?)       ;RESERVED FOR DOS
RECINBLOK   DB      ?               ;RECORD WITHIN BLOCK
RELREC      DW      ?               ;DOUBLE-WORD RELATIVE RECORD NO.
            DW      ?
NORMFCB     ENDS

;*****************************************************************
;
;   DATA SEGMENT
;
;*****************************************************************

DATA        SEGMENT BYTE 'COM'

; Extended FCB No. 1 will be built from command line parameters passed
;   by DOS, and used to look up the sub-directory in its parent.

XFCB1       DB      0FFH,0,0,0,0,0 ;FCB EXTENSION HEADER
            DB      10H             ;ATTR BYTE FOR SUB-DIRS
FCB1        NORMFCB<>               ;NORMAL FCB FOLLOWS EXTENSION

; Extended FCB No. 2 will be built by DOS as a result of lookup in
;   parent directory, and used open & read sub-directory file.

XFCB2       DB      0FFH,0,0,0,0,0,10H
FCB2        NORMFCB<>


; Message strings

DISKERR$    DB      0DH,0AH,'Disk read error',0DH,0AH,'$'
BADFAT$     DB      0DH,0AH,'FAT error: cluster chain not terminated'
CRLF$       DB      0DH,0AH,'$'
COUNT$      DB      ' had'
INREC$      DB      6 DUP(' ')      ;SPACE FOR INPUT REC COUNT DIGITS
            DB      ' clusters, now has'
OUTREC$     DB      6 DUP(' ')      ;SPACE FOR OUTPUT COUNT DIGITS
            DB      0DH,0AH,'$'
ENDMSG$     DB      0DH,0AH
NDIR$       DB      6 DUP(' ')      ;SPACE FOR DIRECTORY COUNT DIGITS
            DB      ' Sub-Directories processed',0DH,0AH
FREED$      DB      6 DUP(' ')      ;SPACE FOR FREED COUNT DIGITS
            DB      ' clusters freed',0DH,0AH,'$'


; Miscellaneous data values

FCODE       DB      11H             ;FUNCTION CODE FOR FIND FIRST
FILECLUS    DW      0               ;FILE SIZE FROM TRACING CLUSTER CHAIN
FATCOUNT    DW      ?               ;NUMBER OF ENTRIES IN FAT
CLUSLEN     DW      ?               ;BYTES PER CLUSTER
SAVSTART    DW      ?               ;SAVE STARTING CLUSTER OF FILE
SECSPER     DB      ?               ;SECTORS PER CLUSTER
RETRY       DB      3               ;3 RETRIES IF READ ERROR
INREC       DW      0               ;CURRENT INPUT RECORD NO.
OUTREC      DW      0               ;CURRENT OUTPUT RECORD NO
NDIRS       DW      0               ;COUNT OF DIRECTORIES PROCESSED
FREED       DW      0               ;COUNT OF CLUSTERS FREED


; Open-ended buffer for holding File Allocation Table

FATWORD     LABEL   WORD            ;PROCESS FAT BY WORDS
DATA        ENDS


;*****************************************************************
;
;   CODE    SEGMENT
;
;*****************************************************************

CODE        SEGMENT BYTE PUBLIC 'COM'

            ORG     80H             ;PARAMATER AREA IN PSP
SUBDATA     LABEL   BYTE            ;USE AS BUFFER FOR 1 SUBDIR ENTRY

            ORG     100H            ;CODE ORIGIN FOR COM PROGRAMS
PACKDIR     PROC    FAR


;*****************************************************************
;   MAINLINE
;*****************************************************************
; Each called routine returns carry flag clear if no errors,
;   set if error.  If error, then DX points to error message.
```

YOUR BUSINESS COMPUTER IS HIDING OUT WITH THE CIA

And there's a good reason for it.

You see, AT&T recently redeveloped the business computer. The new AT&T 3B series is much more compact and easier to use than the old style business computers. The 3B series doesn't need any special computer rooms or air-conditioning systems and it can be placed, ready to go, into nearly any office environment.

But, the one thing that the mighty B's have in common with their ancestors is capability. They all use the AT&T WE™ 32000 microprocessor, the industry's first true 32 bit chip in a system this size. More important, the 3B series is designed to optimize the UNIX™ operating system—which only makes sense, because AT&T invented UNIX.

Now, you've probably guessed that a computer line with this much going for it would only be found at a computer dealer with a lot going for him. That's why your 3B is waiting for you at the CIA.

CIA is Computer Integration Associates. We're one of the few value added resellers nationwide that AT&T has selected to handle the 3B series.

We've got the technical expertise that the 3B's demand, and the support team to back them up. In fact, Computer Integration Associates is an expert at setting up computer systems for businesses of all sizes with applications specialties in accounting, manufacturing, legal, medical/dental and financial systems.

Our PC and Subsystems Division can assist you in upgrading your existing PC systems as well. CIA can provide solutions to all your business computing needs.

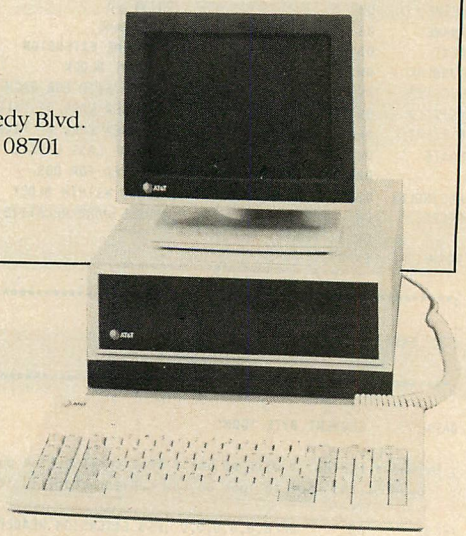
So when you're ready to take a look at the 3B Computer, or upgrade your existing system, you know what you have to do. Look them up at their only local hideout: Computer Integration Associates.

Tell them Joe sent you.

Joe Nowseller

CIA

450 East Kennedy Blvd.
Lakewood, N.J. 08701
(201) 370-3900



AT&T

Computer
Integration
Associates

UNIX is a trademark of Bell Laboratories

```
;   Display the message, then exit to DOS.

            CALL    GETNAME         ;LOOK FOR FILENAME ON COMMAND LINE
            CALL    FATREAD         ;READ IN THE FAT
            JC      MSG
NEXTNAM:    CALL    OPEN            ;LOOK UP FILENAME & OPEN IT
            JC      EOJ             ;ALL DONE WHEN NO MORE MATCHING NAMES
            CALL    DIRSIZE         ;GET FILE SIZE BY TRACING CLUSTERS
            JC      MSG
            CALL    PROCESS         ;PROCESS THE CONTENTS OF THE SUB-DIR
            CALL    CLOSE           ;CLOSE THE SUB-DIR FILE
            JMP     NEXTNAM

EOJ:        MOV     AX,NDIRS        ;CONVERT DIR COUNT TO ASCII
            MOV     SI,OFFSET COM:NDIR$
            CALL    I2ASC
            MOV     AX,FREED        ;CONVERT FREED COUNT
            MOV     SI,OFFSET COM:FREED$
            CALL    I2ASC
            MOV     DX,OFFSET COM:ENDMSG$
MSG:        MOV     AH,9            ;FUNCTION 9 = DISPLAY STRING
            INT     21H             ;CALL DOS TO DISPLAY MESSAGE
            INT     20H             ;RETURN TO DOS
PACKDIR     ENDP


;*****************************************************************
;    PROCEDURE GETNAME
;*****************************************************************
; Test if filename was specified on command line.  If not, build one
;   consisting of all wild characters.  Move name to FCB1, set DTA.

GETNAME     PROC    NEAR
            MOV     SI,5CH              ;POINT TO 1ST FCB IN PREFIX
            CMP     BYTE PTR [SI+1],' ' ;TEST FOR NAME FROM CMD LINE
            JNE     GET1                ;NAME IS THERE
            MOV     CX,11               ;ELSE INSERT 11 WILD CHARS
            LEA     DI,[SI+1]           ;INTO NAME FLD OF FCB1
            MOV     AL,'?'
            REP     STOSB

GET1:       MOV     DI,OFFSET COM:FCB1  ;POINT TO FCB NO. 1
            MOV     CX,12               ;CX = LENGTH OF DRIVE, NAME & EXT
            REP     MOVSB               ;MOVE NAME FROM PREFIX TO FCB 1
            RET
GETNAME     ENDP


;*****************************************************************
;    PROCEDURE FATREAD
;*****************************************************************
; Get the FAT characteristics, calculate the size of the FAT, and read
; the FAT into the buffer.  If read error, reset disk & retry 3 times.

FATREAD     PROC    NEAR
            MOV     DL,FCB2.DRIVE   ;MOVE DRIVE ID INTO DL
            PUSH    DS              ;SAVE THE DATA SEGMENT
            MOV     AH,1CH          ;FUNCTION 1C = GET FAT INFO
            INT     21H             ;CALL DOS
            POP     DS              ;RESTORE DATA SEGMENT
            MOV     SECSPER,AL      ;SAVE SECTORS PER CLUSTER
            MOV     FATCOUNT,DX     ;SAVE NUMBER OF FAT ENTRIES
            CBW                     ;CONVERT SECS PER CLUS TO WORD
            MUL     CX              ;AX=AX*CX, BYTES PER CLUSTER
            MOV     CLUSLEN,AX      ;STORE IT FOR FUTURE USE
            MOV     AX,FATCOUNT     ;RESTORE ENTRY COUNT IN AX
            MOV     DX,AX           ;REPEAT IT IN DX
            INC     DX              ;ADD 1 TO ROUND UP
            SHR     DX,1            ;ENTRY COUNT DIVIDED BY 2
            ADD     AX,DX           ;SIZE IN BYTES = 1.5 * ENTRY COUNT
            ADD     AX,CX           ;ROUND UP TO NEXT SECTOR
            DEC     AX              ;  BY ADDING SECTOR SIZE LESS 1
            CWD                     ;CONVERT TO DBL WORD IN DX:AX
            DIV     CX              ;AX / CX = FAT LENGTH IN SECTORS
            MOV     CX,AX           ;MOVE SECTOR COUNT TO CX

            MOV     AH,19H          ;DETERMINE DEFAULT DRIVE
            INT     21H             ;RETURNS DRIVE ID IN AL
            MOV     DX,1            ;BEGIN AT SECTOR 1
            MOV     BX,OFFSET COM:FATWORD   ;POINT TO BUFFER FOR FAT
F1:         PUSH    AX              ;SAVE THE REGS,
            PUSH    BX              ;   BECAUSE INT 25
            PUSH    CX              ;   DESTROYS THEM
            PUSH    DX
            INT     25H             ;ABSOLUTE DISK READ
            POP     DX              ;RESTORE REGS EXCEPT AX
            POP     CX
            POP     BX
            JNC     F2              ;JUMP IF NO DISK ERRORS
            POPF                    ;POP FLAGS SAVED BY INT 25
            SUB     AH,AH           ;AH=0 IS DISK RESET FUNCTION
            INT     13H             ;DISK I/O INTERRUPT
            POP     AX              ;RESTORE SECTOR COUNT
            DEC     RETRY           ;DECREMENT RETRY COUNT
            JNZ     F1              ;TRY READING AGAIN IF NOT ZERO
            MOV     DX,OFFSET COM:DISKERR$  ;ERROR EXIT
            STC
            RET

F2:         POPF                    ;READ OK: RESTORE STACK
            POP     AX
            CLC
            RET
FATREAD     ENDP


;*****************************************************************
;    PROCEDURE OPEN - find sub-dir name & open it as a data file
;*****************************************************************
; Input: AH is 11 to find first name, 12 to find subsequent.
;   If found, but not a sub-directory, look for next matching entry,
;   until a matching sub-directory is found or there are no more
;   matching entries.  If no match, exit with carry flag on,
;   else open the file, save starting cluster pointer,
;   display its name & turn carry flag off.

OPEN        PROC    NEAR
            MOV     DX,OFFSET COM:XFCB2 ;SET DTA AT EXTENDED FCB 2
            MOV     AH,1AH              ;FUNCTION 1A = SET DTA
            INT     21H                 ;CALL DOS TO SET DTA
            MOV     DX,OFFSET COM:XFCB1 ;POINT TO EXTENDED FCB NO. 1

OP1:        MOV     AH,FCODE            ;GET CODE 11 OR 12
            INT     21H                 ;CALL DOS TO FIND FIRST/NEXT
            CMP     AL,0FFH             ;AL=FF IF NOT FOUND
            JNE     OP2                 ;SKIP IF FOUND
            STC                         ;SET CARRY FLAG IF NOT FOUND
            RET

OP2:        MOV     FCODE,12H                   ;CODE 12 = FIND NEXT
            CMP     BYTE PTR FCB2+1.ATTR,10H    ;IF FOUND, TEST IF A SUB-DIR
            JNE     OP1                         ;IF NOT, GO FIND NEXT ENTRY
            CMP     BYTE PTR FCB2+1.DNAME,'.';IF SUB-DIR, TEST IF PERIOD
            JE      OP1                         ;IF SO, GO FIND NEXT ENTRY

            MOV     AX,FCB2+1.STARTCL   ;GET STARTING CLUSTER FROM FCB
            MOV     SAVSTART,AX         ;SAVE IT FOR FUTURE USE
            MOV     DX,OFFSET COM:XFCB2 ;POINT TO EXTENDED FCB
            MOV     AH,0FH              ;OPEN FILE FUNCTION
            INT     21H
            MOV     SI,OFFSET COM:FCB2.FNAME ;POINT TO NAME IN OPEN FCB
            MOV     CX,11               ;DISPLAY 11 CHARS OF FILENAME
            MOV     AH,2                ;DISPLAY CHARACTER FUNCTION

OP3:        MOV     DL,[SI]             ;GET FILENAME CHARACTER
            INT     21H                 ;DISPLAY IT
            INC     SI                  ;POINT TO NEXT CHAR IN NAME
            LOOP    OP3

            CLC                         ;INDICATE NO ERRORS
            RET
OPEN        ENDP


;*****************************************************************
;    PROCEDURE DIRSIZE
;*****************************************************************
; Trace thru FAT, counting clusters in allocation chain.
;   For each cluster, add bytes per cluster to file size.
;   At exit, file size in bytes is stored in open FCB, file size
;   in clusters in location FILECLUS.
;   Register usage:  DX:AX is dword accumulator for file size
;                    BX is last cluster number
;                    CX contains count of FAT entries
;                    SI points to next FAT entry
;                    DI contains bytes per cluster

DIRSIZE     PROC    NEAR
            MOV     BX,SAVSTART     ;PUT STARTING CLUSTER IN BX
            MOV     DI,CLUSLEN      ;BYTES PER CLUSTER IN DI
            MOV     CX,FATCOUNT     ;LOOP COUNT IN CX
            SUB     AX,AX           ;ZERO OUT FILE SIZE
            MOV     DX,AX           ;AND CLUSTER COUNT
            MOV     FILECLUS,AX     ;AND CLUSTER COUNT
```

HARD DISKS ARE NOT LIKE POLITICS, THERE IS NOTHING CONFUSING ABOUT THEM....

....AT LEAST NOT AT THIS CIA.

SPECIAL NEW YEAR'S SALE ON ALL INTERNAL SUBSYSTEMS!

Reliance 10 Megabyte Internal	Was $1,165	New Year's Special:	$ 875
Reliance 10 Megabyte External			$1,265
Reliance 10 Megabyte External w/3.3 MB Floppy Backup			$1,995
Reliance 20 Megabyte Internal	Was $1,650	New Year's Special:	$1,295
Reliance 20 Megabyte External			$1,750
Reliance 20 Megabyte External w/3.3 MB Floppy Backup			$2,499
Reliance 40 Megabyte Internal	Was $2,270	New Year's Special:	$1,775
Reliance 40 Megabyte External			$2,370
Reliance 40 Megabyte External w/3.3 MB Floppy Backup			$3,120
Reliance 3.3 MB Floppy Internal			$ 875
Reliance 3.3 MB Floppy External			$1,250

ALL DRIVES INCLUDE CONTROLLER CARDS AND CABLES

ALL SUBSYSTEMS ALSO AVAILABLE AS PART OF AT&T PC 6300 CONFIGURATION

Dealer inquiries invited

Volume discounts available

Call for technical specifications, price lists or information on other CIA computer products and services.

CIA Computer Integration Associates

450 East Kennedy Blvd.
Lakewood, N.J. 08701
(201) 370-3900

Free CIA's unmatched service

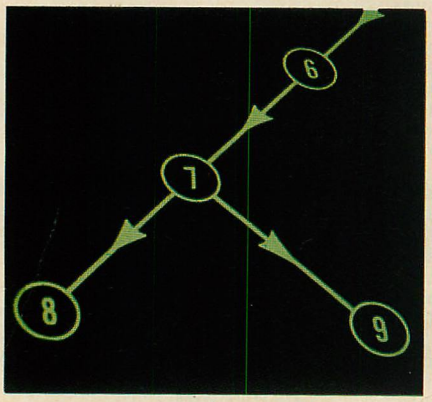
Free one year guarantee

CIRCLE NO. 209 ON READER SERVICE CARD

```
D1:     ADD     AX,DI           ;UPDATE FILE LENGTH
        ADC     DX,0            ;CARRY INTO HI WORD
        INC     FILECLUS        ;UPDATE CLUSTER COUNT
        MOV     SI,BX           ;CALC BX*1.5 IN SI
        SHR     BX,1
        PUSHF                   ;SAVE THE CARRY FLAG
        ADD     SI,BX
        MOV     BX,FATWORD[SI] ;GET NEXT FAT ENTRY
        POPF                    ;RESTORE CARRY FROM SHIFT
        JNC     D2              ;SKIP IF BX WAS EVEN
        SHR     BX,1            ;IF ODD, RIGHT-JUST
        SHR     BX,1            ;  HI 12 BITS OF CLUSTER NO.
        SHR     BX,1
        SHR     BX,1
D2:     AND     BX,0FFFH        ;ZERO OUT 4 HI BITS
        CMP     BX,0FF8H        ;TEST FOR END OF CHAIN
        JGE     D3              ;END IF CLUSTER IS FF8 OR ABOVE
        LOOP    D1              ;LOOP IF NOT

        MOV     DX,OFFSET COM:BADFAT$    ;IF FAT END BUT NOT CHAIN END,
        STC                              ;    THEN ERROR
        RET

D3:     MOV     FCB2.FSIZELO,AX     ;PUT FILE LENGTH INTO FCB
        MOV     FCB2.FSIZEHI,DX
        CLC                         ;CLEAR ERROR FLAG
        RET
DIRSIZE ENDP


;*****************************************************************
;    PROCEDURE PROCESS
;*****************************************************************
; Read the sub-directory as a data file, 1 entry at a time.
;   Write out entries ehich are not deleted.  At first unused entry,
;   write a zero-filled entry and truncate the file to that length.
;   Regs:  SI points to I/O buffer.

PROCESS PROC    NEAR
        SUB     AX,AX
        MOV     INREC,AX
        MOV     OUTREC,AX
        MOV     SI,OFFSET COM:SUBDATA;POINT TO I/O BUFFER
        MOV     DX,SI                ;SET DTA TO BUFFER
        MOV     AH,1AH
        INT     21H
        MOV     FCB2.RECSIZE,32     ;INSERT REC SIZE INTO FCB
        MOV     DX,OFFSET COM:XFCB2 ;POINT TO EXTENDED FCB

READ:   MOV     AX,INREC        ;GET RECORD (ENTRY) NO. TO READ
        MOV     FCB2.RELREC,AX ;PUT IT INTO FCB'S RELATIVE REC FLD
        INC     INREC           ;UPDATE RECORD NO. FOR NEXT READ
        MOV     AH,21H          ;PERFORM DIRECT ACCESS READ
        INT     21H
        TEST    AL,AL           ;EOF IF AL NOT ZERO
        JNZ     EOF

        CMP     BYTE PTR SUBDATA,0E5H    ;IS ENTRY DELETED?
        JE      READ            ;YES: DO NOT WRITE, GO READ NEXT

WRITE:  MOV     AX,OUTREC       ;GET RECORD NO. TO WRITE
        MOV     FCB2.RELREC,AX ;PUT IT INTO FCB
        INC     OUTREC          ;UPDATE REC NO. FOR NEXT READ
        MOV     AH,22H          ;PERFORM DIRECT ACCESS WRITE
        INT     21H
        CMP     BYTE PTR SUBDATA,0 ;NEVER-USED ENTRY?
        JNE     READ            ;IF NOT, READ NEXT, ELSE EOF

EOF:    RET                     ;RETURN AT END OF DATA
PROCESS ENDP


;*****************************************************************
;   PROCEDURE CLOSE: Close the sub-dir file, display counts
;*****************************************************************
CLOSE   PROC    NEAR
        MOV     AX,OUTREC       ;GET COUNT OF ENTIRES WRITTEN
        TEST    AX,AX           ;MAKE SURE SOME WERE WRITTEN
        JZ      CL1
        MOV     FCB2.RELREC,AX ;SET RECORD COUNT IN FCB
        SUB     CX,CX           ;WRITE NOTHING, JUST SET SIZE
        MOV     DX,OFFSET COM:XFCB2 ;POINT TO EXTENDED FCB
        MOV     AH,28H          ;PERFORM BLOCK WRITE, SET SIZE
        INT     21H

CL1:    MOV     AX,FILECLUS     ;GET INPUT CLUSTER COUNT
        ADD     FREED,AX        ;UPDATE FREED CLUSTER COUNT
        MOV     SI,OFFSET COM:INREC$
        CALL    I2ASC           ;CONVERT COUNT TO ASCII DIGITS

        MOV     AX,OUTREC       ;GET OUTPUT RECORD COUNT
        MOV     BX,32           ;BYTES PER ENTRY
        MUL     BX              ;DX:AX = OUTPUT BYTES
        DIV     CLUSLEN         ;GET OUTPUT CLUSTERS, REMDR IN DX
        TEST    DX,DX           ;TEST FOR REMAINDER
        JZ      CL2
        INC     AX              ;IF REMDR, ROUND UP TO NEXT CLUSTER
CL2:    SUB     FREED,AX        ;TOTAL FREED = FREED + INPUT - OUTPUT
        MOV     SI,OFFSET COM:OUTREC$
        CALL    I2ASC           ;CONVERT OUTPUT COUNT TO DIGITS
        MOV     DX,OFFSET COM:COUNT$     ;DISPLAY COUNT MSG
        MOV     AH,9
        INT     21H
        INC     NDIRS           ;INCREMENT DIRECTORY COUNT
        RET
CLOSE   ENDP
CODE    ENDS


COMMENT " ***************************************************************
 MODULE I2ASC CONVERTS 2-BYTE INTEGER INTO 6-BYTE NUMERIC ASCII STRING
     INPUT:  NUMBER TO BE CONVERTED IN AX
             DS:SI POINTS TO STRING TO RECEIVE OUTPUT
     OUTPUT: STRING AT DS:SI, RIGHT JUSTIFIED, BLANK PADDED
                  IF AX=0, STRING IS 1 ZERO IN RIGHTMOST POSITION
                  IF AX<0, LEADING MINUS IS FLOATED BEFORE 1ST DIGIT
             DS:DI POINTS TO FIRST NON-BLANK IN STRING
             ALL OTHER REGISTERS UNCHANGED
*****************************************************************************
"
CODE    SEGMENT BYTE PUBLIC 'COM'
        ASSUME  CS:COM
I2ASC   PROC    NEAR
        PUBLIC  I2ASC
        PUSH    ES              ;SAVE REGISTERS
        PUSH    DX
        PUSH    CX
        PUSH    BX
        PUSH    AX

        MOV     AL,' '          ;BLANK OUT STRING
        MOV     DX,DS
        MOV     ES,DX
        MOV     DI,SI
        CLD
        MOV     CX,6
        REP STOSB

        MOV     DI,SI
        ADD     DI,5            ;POINT AT STRING END
        MOV     BX,10           ;BASE TEN DIVISOR
        POP     AX              ;GET BINARY NUMBER INTO AX
        PUSH    AX
        STD                     ;MOVE BACKWARDS THRU STRING

DIVLOOP: CWD                    ;CONVERT TO DBL WORD IN DX,AX
        IDIV    BX              ;QUOTIENT IN AX, REMAINDER IN DX
        TEST    DL,80H          ;TEST IF REMAINDER NEGATIVE
        JZ      $+4             ;SKIP IF POS
        NEG     DL              ;ELSE GET ABS OF REM
        OR      DX,30H          ;INSERT ASCII ZONE
        XCHG    AX,DX           ;EXCHANGE QUOT & REM
        STOSB                   ;STORE ASCII CHAR IN STRING
        MOV     AX,DX           ;RESTORE QUOTIENT
        TEST    AX,0FFFFH       ;TEST IF MORE DEC DIGITS
        JNZ     DIVLOOP

        POP     BX              ;RESTORE INTEGER INTO BX
        TEST    BX,8000H        ;TEST IF NUMBER NEG
        JZ      I2EXIT
        MOV     AL,'-'          ;INSERT MINUS SIGN IF NEG
        STOSB

I2EXIT: INC     DI              ;POINT AT LAST NON-BLANK
        MOV     AX,BX           ;RESTORE REGS
        POP     BX
        POP     CX
        POP     DX
        POP     ES
        CLD
        RET
I2ASC   ENDP

CODE    ENDS
        END     PACKDIR
```

Tree Structures

A tutorial on using tree structures for random data storage and retrieval.



ATINDRA CHATURVEDI

This is the first of two articles on B-tree-based file management systems. In it, the advantages and disadvantages of tree structures are discussed. The second part, to appear next month, will describe a file management system based on the B+tree structure and will include the source code program in C language to implement the system and its associated data access system.

The problem of organizing data so that they can be quickly stored, updated, and retrieved has plagued computer scientists for years. "Tree" structures have shown themselves to be an efficient solution.

A *tree* consists of nodes and branches between nodes. The node corresponding to the start of a linked list is called the *root node*. The branches correspond to the pointers (links) of the list. Nodes with no pointers are called *leaves*. Figure 1 shows how a linked list can be represented as a tree. To locate an element in a tree, start at the root and traverse the tree using the pointers in the root to the lower nodes, and so on, until the desired amount is found.

In a generalized tree structure as in figure 1, a multiway branching decision has to be taken at each node, since there may be more than one pointer to follow. Choosing a pointer to follow at random may lead to inefficiencies if the desired element cannot be found in that path; the only alternative would be to retrace the path and make another attempt using a different pointer.

The *binary search tree* in figure 2 solves the problem neatly by allowing only two pointers in any node (except the leaves) and only one key. One of the pointers is restricted to pointing to a node with a lower key value (if one exists), relative to the current node. The other pointer is allowed to point only to a node with a higher key value.

These restrictions allow more efficient searching of the binary search tree: only a two-way decision has to be taken at any node. Deciding which of these two paths to take depends on whether the value of the key being searched for has a lower or higher key value than the current node.

The key does not exist in the tree under one of the following two conditions: the key being searched for has a higher key value than the node key, but the pointer to the higher node is null; or the key being searched for has a lower key value than the node key, and the pointer to the lower node is null.

The above procedures are more suited to searching for data stored in internal memory than on external storage devices because of the extensive I/O operations required to traverse the file. A binary search tree requires an aver-

PHOTOGRAPH • MARC COHEN


1
2
3
4
5
6
7
8
9

age of $\log_2 N$ comparisons before locating the specified key (see table 1).

The efficiency of data retrieval in a binary search tree is totally dependent on the order of the keys when the binary search tree is constructed. In the most extreme case, a degenerate binary search tree results if the keys arrive in sorted ascending or descending order. This results in a binary search tree that is equivalent to a linearly linked list or in an unbalanced binary search tree, leading to key retrieval times that are highly variable (see figure 3).

Many algorithms have been published that would solve the problem of maintaining a well-balanced tree at all times, but they are sufficiently complex so as to preclude widespread use.

B-TREE STRUCTURES

A new approach to searching for data by way of multiway tree branching (rather than two-way branching, as in a binary search tree), was published in 1970 by R. Bayer and E. McCreight. This was based on a new type of data structure called a B-tree, which allowed insertion and retrieval of data by key in large files using simple algorithms.

In a B-tree structure, data records are stored in each node so that by traversing the tree the desired record may be found in any node of the tree.

A *B-tree of order n* is defined as having the following properties:

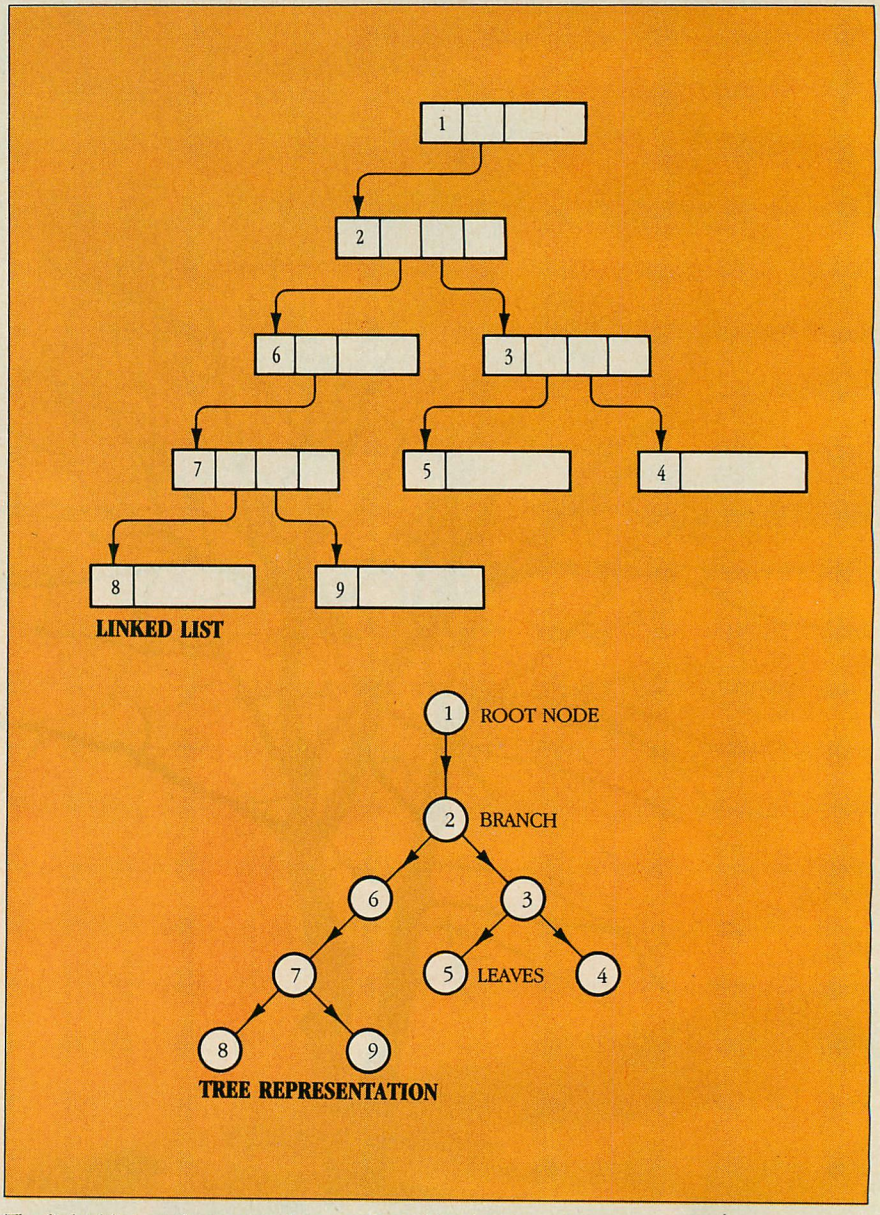
- Every node has no more than n immediate successors.
- Every node, except for the root and the terminal nodes, has at least $n/2$ immediate successors.
- The root has either none or at least two successors.
- All leaf nodes are at the same level and they do not contain any pointers to succeeding nodes.
- A nonleaf node with k immediate successors has k-1 keys.

Figure 4 shows a B-tree of order 4. In each node a maximum of three keys can be stored. The root node, in this case, has two successors.

Search procedure for a B-tree. To search a B-tree for a particular key, begin at the root node and compare the keys in the root with the desired key. If any key matches, the search is successful. If no match is found, select the appropriate pointer and compare with the keys in the newly selected node. Continue this until a terminal node is reached or the pointer is null, in which case the record is not in the B-tree.

Insertion procedure for a B-tree. When a record with key 400 is to be inserted into the B-tree of figure 4, the following events take place:

FIGURE 1: *Representation of Linked List as a Tree*



The linked list on the top can be represented by the generalized tree on the bottom.

- The leaf node for the new record is located (middle of figure 4).
- The new record is inserted, resulting in the B-tree shown at the bottom of figure 4.
- As the node in which the new record was inserted (node 3) still has three keys (one key less than the order of the B-tree), the insertion procedure is complete.

The resulting B-tree receives a record with key 450 in figure 5:

- The leaf node for the new record is located (top of figure 5).
- The new record cannot be inserted into the node because the number of keys in the node would be equal to the order of the B-tree. Thus, node 3 has to be split into another node to accommodate the new record.
- Node 5 is created to hold the overflow from node 3. Records 450 and 500 are moved to node 5. Record 400 is moved up one level and a place to insert it is found in that node (in this case, the root node). Its previous location in node 3 is cleared.
- Record 600 is moved over one position along with its corresponding pointer. Record 400 is inserted in the root node. The record number of the new node is inserted in the root node (bottom of figure 5).

The B-tree now satisfies all the rules; the insertion procedure is complete.

Searching a B-tree of order n is more efficient than a binary search tree because a B-tree allows a multiway de-

LET'S TALK EXPERT-TO-EXPERT

PC TECH JOURNAL, the magazine written for sophisticated IBM PC users...talks with you expert to expert!

Do you require a magazine that provides you with the insight and knowledge to act as your silent partner when designing new systems? Have you needed to consult with authorities in telecommunications to resolve a nagging problem? Wouldn't a second opinion about connecting your PC to a main-frame be helpful? With your subscription to PC TECH JOURNAL, you're harnessing experts in your field...experts who will talk to you in the language you understand, about the concerns you have...EXPERT TO EXPERT!

PC TECH JOURNAL is the technologically sophisticated magazine written for experts in the field of personal computing like yourself...discussing the concerns experts have...developing elegant programming methodologies only experts can understand...covering the whole field of IBM PC's with thought-provoking articles on communications, distributed processing, office automation, networking and programming.

Subscribe today and save up to 27% off the full subscription price, and have PC TECH JOURNAL delivered to your home or office every month! From One Expert to Another: *Subscribe Today!*

TECH PC JOURNAL P.O. Box 2966, Boulder, Colorado 80322

PT4Z098

YES, I want to communicate with other experts and professionals about IBM PC's and compatible machines! Please enter my subscription to PC TECH JOURNAL for:

- ☐ 8 issues for $17.97—SAVE 10%!
- ☐ 12 issues for $24.97—SAVE 17%!
- ☐ 24 issues for $43.97—SAVE 27%!

Mr./Mrs./Ms._____
please print name in full

Company_____

Address_____

City_____ State_____ Zip_____

Savings based on full one-year (12 issues) subscription price of $29.97.

Check one: ☐ Payment enclosed. ☐ Bill me later.

Charge my: ☐ American Express ☐ Visa ☐ MasterCard

Card No._____ Exp. Date_____

Please add $1 per issue in Canada and all other foreign countries. Please allow 30 to 60 days for delivery of first issue.

Not long ago, *PC* Magazine called MDBS III "The most complete and flexible data base management system available for microcomputers." That's a powerful statement. But then, MDBS III is an amazingly powerful software package. So powerful, in fact, that it lets you build mainframe-quality application systems on your micro or mini. MDBS III is not for beginners. It's for application developers with large data bases or complex data interrelationships who want to define data base structures in the most natural way—without resorting to redundancy or artificial constructs. It's for professionals who can appreciate its extensive data security and integrity features, transaction logging, ad hoc query and report writing capability and its ability to serve multiple simultaneous users. And if you want the power and the glory that only the world's most advanced data management system can provide, MDBS III is for you. For information on MDBS III and our professional consulting services, write or call Micro Data Base Systems, Inc., MDBS/Application Development Products, 85 West Algonquin Road, Suite 400, Arlington Heights, IL 60005. (800) 323-3629, or (312) 981-9200. **MDBS III. ABSOLUTE POWER.**

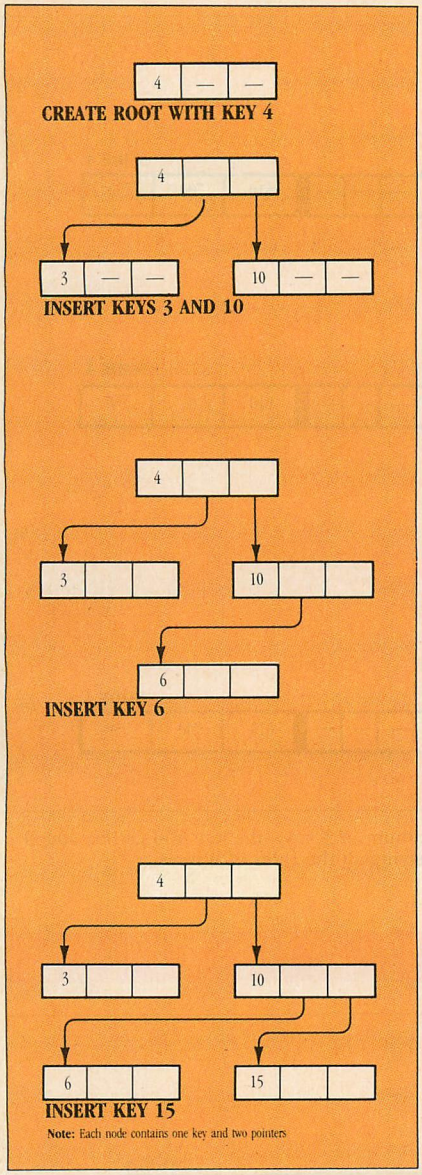
WE'LL GIVE YOU THE POWER.

YOU TAKE THE GLORY.

MDBS III is a trademark of Micro Data Base Systems, Inc.

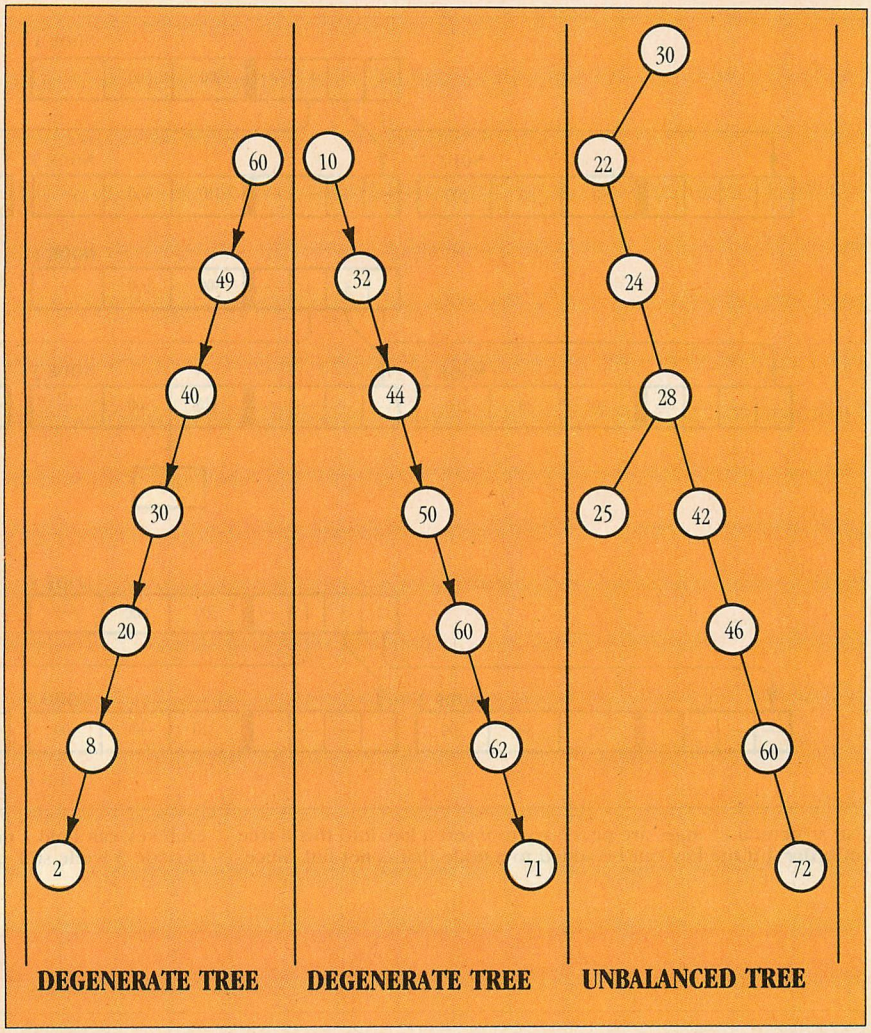
CIRCLE NO. 219 ON READER SERVICE CARD

FIGURE 2: *Binary Search Tree*



The creation of a binary tree shows insertion of keys with pointers are correctly set.

FIGURE 3: *Degenerate and Unbalanced Trees*



Degenerate or unbalanced binary trees can be caused by the arrival order of keys. This results in binary search trees with highly variable key retrieval times as shown in the three structures above.

cision at each node. In a binary search only half of the remaining keys are eliminated at each step. Consider, however, that in a B-tree of order n, approximately $(n\text{-}1)/n$ of the remaining keys are eliminated at each step (refer to table 1).

A binary search tree tends to be long and narrow with many levels and few nodes at each level. The B-tree tends to be short and wide with few levels and many nodes at each level. The search time depends on the depth of the tree, which tends to be smaller in a B-tree structure.

B+TREE STRUCTURE

In the B+tree structure, only keys and pointers to other nodes (not entire records) are stored in the nonleaf nodes, so more keys can be stored in a node of a given size. Thus, more paths are available for branching in a node. This leads to shorter search times as fewer seeks have to performed to locate a given key, giving the B+tree an advantage over a conventional B-tree. For a comparison of how the order of a B+tree affects the number of seeks needed to locate a key, see table 1.

However, each search for a given key has to start at the root node and access all levels of the B+tree. Unlike a B-tree, in which a record could be found at an intermediate level, pointers to data records in a B+tree are available only in the leaf nodes.

Insertion procedure for a B+tree. A B+tree of order 3 is illustrated in figure 6. The file shown in this figure contains records for characteristics of houseplants; keyed access to this file using the plant name as the key is desired.

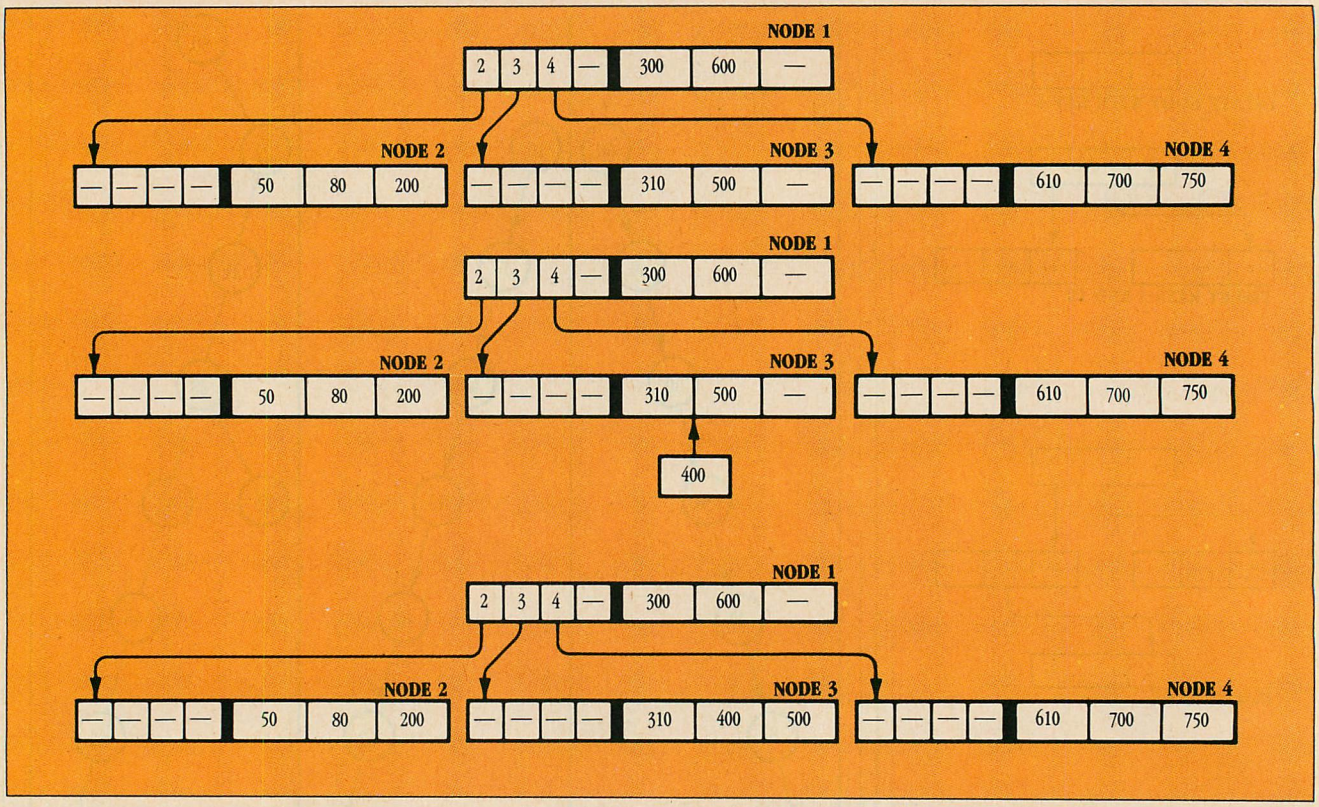
The structure of a node in the B+ tree is shown at the top of figure 6. Since the B+ tree is of order 3, a node can contain only three pointers and two keys. If the node contains pointers to data records, the flag will have value *Y*; otherwise the value will be *N*. A NULL value in a pointer means it is not pointing to anything. A null string in a key means the key area is unused.

The following insertion procedure and its implementation (see figure 6) follows the description in the following works: Lewis's *Pascal Programming for the Apple* and Lewis and Smith's *Applying Data Structures.*

Insert record #1 "Fuchsia": The key Fuchsia is inserted along with the data pointer 1. Since the node contains pointers to data records, the value of the flag in the node is *Y*.

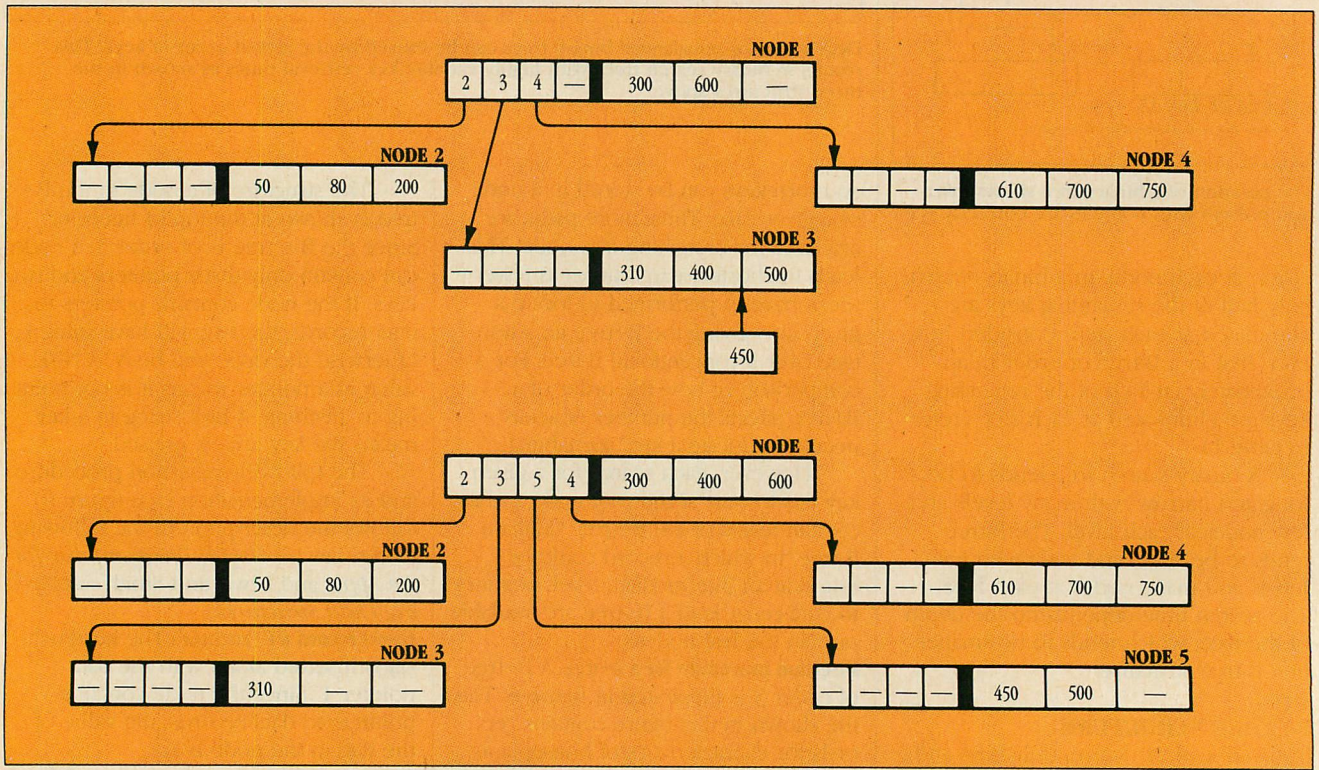
Insert record #2 "Croton": The key Croton is inserted in the node such that the

FIGURE 4: *Insertion into B-tree without Node Split*



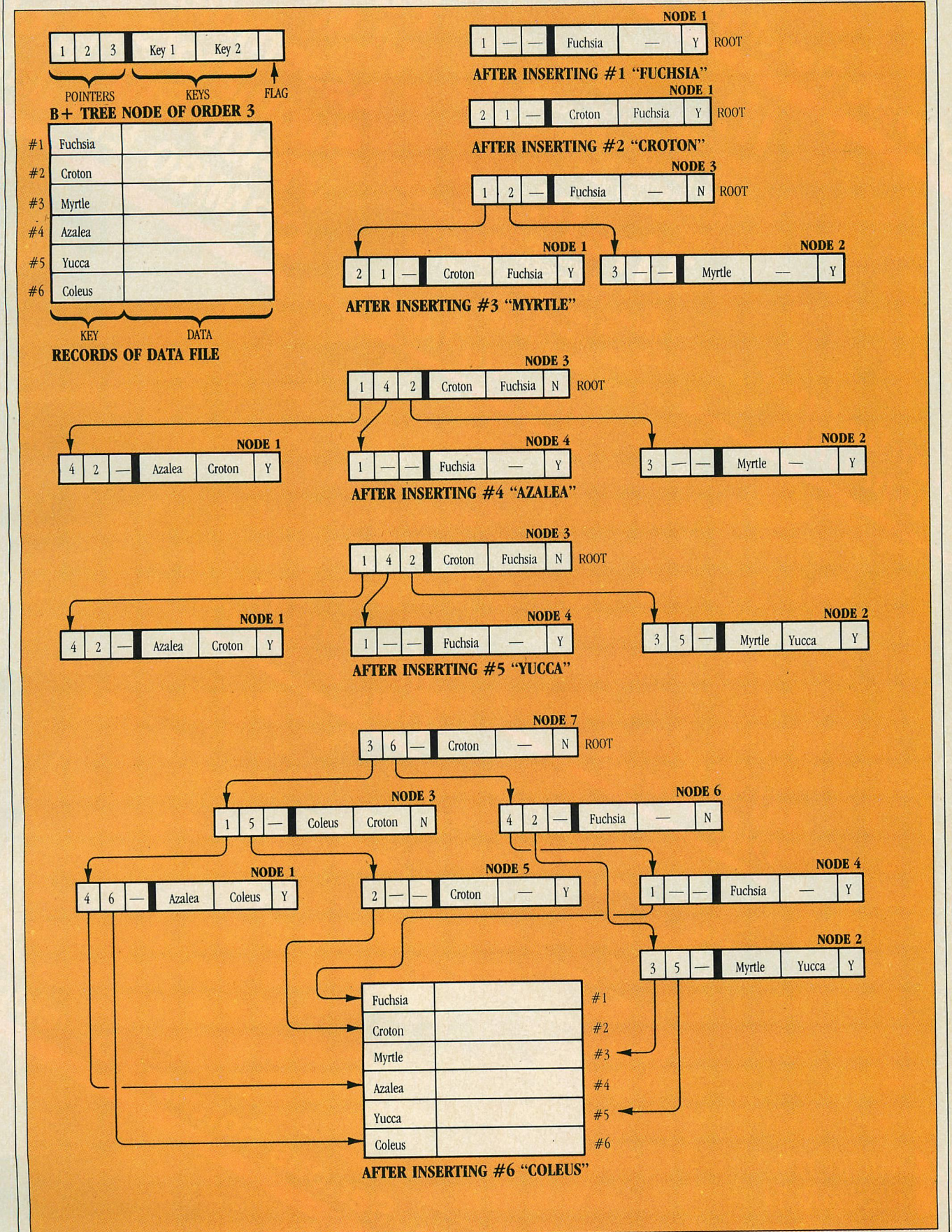
No structural changes are necessary to insert a key into this B-tree of order 4 if the key can be added to a node that is not full. Since each key can hold a maximum of 3 keys, the key 400 can be added to node 3 while still satisfying all the rules of B-trees.

FIGURE 5: *Insertion into B-tree with Node Split*



To add a key to a node that already contains the maximum number of keys, the node must be split and the pointers readjusted. Thus to insert key 450, node 5 is created to hold the overflow from node 3 and the pointers on the level above are changed.

FIGURE 6: *Insertion into B+Tree*



A B+tree of order 3 that indexes a file containing data on houseplants is constructed step by step. Since the B+tree is of order 3, any node can hold three pointers and two keys. The flag indicates if the pointers in the node point to data records or other nodes.

Software Development Tools

C
Pascal
PLM
Assemblers
Simulators
VT-100 Emulator
CPM/80 Emulator

If:

Manufacturer-Compatible, Fully-Supported Cost-Effective, Cross and Native Development tools are important to you!

Then:

Let us tell you about MICROTEC RESEARCH's extensive line of field-proven software development tools for serious software developers.

Manufacturer Compatible

MICROTEC RESEARCH has been providing flexible and economical solutions for software developers since 1974. We've grown with the industry as our cross software development tools have transformed many large and small computers into powerful cross development systems for microprocessor applications. Our software tools are manufacturer compatible. Therefore, software made using manufacturers development systems may be more productively used in the MICROTEC RESEARCH cross-development environment.

Fully-Supported

MICROTEC RESEARCH's products are continually maintained and updated based upon the experience of thousands of installations worldwide. Revisions and modifications are distributed to licensees in warranty or covered by a service contract. Upgrades, which are major enhancements to a product's capabilities, are available at a significant discount. Our update/upgrade policy assures users they'll always be current with the fast moving world of microprocessor development.

Effective Differences

Beginning with product concept, through development, quality assurance, and post-sales support — **Quality, Compatibility,** and **Service** are the differences which set MICROTEC RESEARCH apart from the others.

Start Saving Time & Money

If you are a serious software developer, shopping for software development tools, call or write today for more information. **800-551-5554** In CA call **(408) 733-2919**

Target Microprocessors	Host Computers	Software Tools
8086/80186	DEC VAX, PDP-11	C Compilers
8096	DG MV-Series	Pascal Compilers
8080/8085	DG Eclipse	PLM Compiler
8051	Apollo	Assemblers
8048	UNIX Systems	Linking Loaders
Z8002	IBM PC	Librarians
Z80	Data General/ONE	Download
Z8	HP 150	Communications
NSC 800	DEC Rainbow	VT-100 Emulator
9900...	others	CPM/80 Emulator
68000/08/10		
6809		
6800/01/02		
6805		
6301		
6305		
6500		
G65SCXX, G65SC1XX		
R65C00, R6500/1..		
others		

for Serious Software Developers

MICROTEC is a trademark of Microtec Research, Inc. Santa Clara, CA

MICROTEC™ RESEARCH

3930 Freedom Circle, Suite 101, Santa Clara, CA 95054
Mailing Address: P.O. Box 60337, Sunnyvale, CA 94088
(408) 733-2919 • Telex (ITT) 4990808

CIRCLE NO. 143 ON READER SERVICE CARD

keys in the node are in ascending sequence. The key Fuchsia and its data pointer 1 are shifted to the right to make room for the new key and its data pointer. The value of the flag is still *Y*.

Insert record #3 "Myrtle": The key Myrtle should be inserted after key Fuchsia but doing so would overflow the node. So the root node #1 is split, and node #2 is created to hold the overflow. The key Myrtle and its data pointer 3 are inserted into node #2. A new root node #3 is created. It holds the "middle" key of the split node and a pointer to the original node. The pointer to the new node #2 is also inserted in the new root. Since the pointers in the root point only to nodes in the tree, the value of the flag is *N*. The value of the flag in node #1 and node #2 is *Y*.

Insert record #4 "Azalea": The key Azalea belongs in leaf node #1, so the keys Croton and Fuchsia, along with their pointers, are shifted to the right to make room. This leads to node overflow, so node #1 is split and node #4 is created to hold the overflow from node #1, which is the key Fuchsia and its data pointer 1. The key Azalea and its data pointer 4 are inserted into node #1. The key and pointer areas in node #1 vacated by Azalea and its data pointer are cleared. Node #3 contains pointers to data so the value of the flag is set to *Y*. The key of the new node #3 Fuchsia and the number of its node are inserted in the root node.

Insert record #5 "Yucca": The key Yucca belongs in node #2 after the key Myrtle. Insertion of the key Yucca and its data pointer in node #2 does not lead to overflow in the node, so the insert procedure ends.

Insert record #6 "Coleus": The key Coleus belongs in node #1 between Azalea and Croton. The key Croton and its pointer are shifted to the right by one place, leading to overflow in the node. Node #5 is created to hold the overflow key Croton and its data pointer. The pointer in this node is to a data record so the value of the flag is *Y*.

The key and pointer areas in node #1 vacated by Croton are cleared. The key of the new node #5 Croton is inserted into the node at the next higher level (where it already exists). The pointer for key Croton in node #3 is updated from node #1 to node #4.

The "middle" key from node #1 is then moved up to the node at the next higher level. This leads to overflow of root node #3. A new node #6 is created to hold the overflow key Fuchsia and its associated pointer. The new node #6 has value *N* in the flag since the pointer in it points to a node in the tree. The key Coleus and its node pointer are inserted into node #3.

Since the root node was split, a new root node #7 is created. This contains the "middle" key of node #3 and the node pointer 3 and a pointer to the lower node #5. In this case, this insertion does not lead to any overflow, so the insertion procedure ends.

The **btsys** program (which will be described in detail in part two) is loaded into memory and stays resident as an extension of DOS. It is possible to access this system from any user program written in any language, as long as it adheres to the interface protocol required by the file management system function **btsys**.

The functions provided by **btsys** are requested through user interrupts. Therefore, the user program does not have to be linked to the file management code. An interface routine in the C language is provided to shield the user program from having to deal with the details of interrupts and other machine-dependent particulars.

TABLE 1: *Number of Seeks on File to Locate a Key*

NUMBER OF RECORDS IN FILE	NUMBER OF COMPARISONS					
	LINEAR SEARCH[1]	BINARY SEARCH[1]	B−TREE[2] ORDER 10	B−TREE[2] ORDER 100	B+TREE[3] ORDER 10	B+TREE[3] ORDER 100
10	5	4	3	2	1	1
100	50	7	4	2	2	1
1,000	500	10	5	3	3	2
10,000	5,000	14	7	4	4	2
100,000	50,000	17	8	4	5	3

[1]Average number of seeks
[2]Maximum number of seeks
[3]Fixed number of seeks

If N = number of records in data file
n = order of tree structure
K = number of seeks, then

For LINEAR SEARCH $K = \frac{N}{2}$
BINARY SEARCH $K = \log_2 N$
B−tree $K \leq 1 + \log\left\lceil \frac{n}{2} \right\rceil \frac{N+1}{2}$
B+tree $K = \lceil \log_n N \rceil$

In a binary search, only half the remaining keys are eliminated at each step. In a B-tree of order *n*, approximately *(n-1)/n* of the remaining keys are eliminated at each step.

REFERENCES

Adel'son-Vel'ski, G.M., and E.M. Landis. *Soviet Math. 3*: 1,259-63.

Bayer, R. Proc ACM-SIGFIDET Workshop (1971):219-35.

Bayer, R., and E. McCreight. *Acta Informatica* (1972):173-89.

Burge, W.H. "Sorting, Trees, and Measures of Order," *Information and Control* 1 (1958):181-97.

Clampett, H.A. "Randomized Binary Searching with Tree Structures," *Comm ACM* 7 (1964):163-65.

Foster, C.C. "A Study of AVL Trees," *Report GER-12158*. Goodyear Aerospace Corp. (April 1965).

Foster, C.C. *CACM 16* (1973):513-17.

Ghosh, S.P., and M.E. Sanko. *Journal ACM, 16* (1969):569-79.

Horowitz, E., and S. Sahni. *Fundamentals of Data Structures*. Rockville: Computer Science Press, 1976.

Hu, T.C., and A.C. Tucker. *SIAM J. Applied Math, 21* (1971):514-32.

Kernighan, B., and D. Ritchie. *The C Programming Language*. Englewood Cliffs: Prentice-Hall, 1978.

Knuth, D.E. *Acta Informatica* 1 (1971):14-25, 270.

Knuth, D.E. *The Art of Computer Programming*. Vol. 3. Reading: Addison-Wesley Publishing Co., 1973, 422-79.

Kochan, S.G. *Programming in C*. Rochelle Park: Hayden Book Co., 1984.

Landauer, W.I. *IEEE Trans. EC-12* (1963):863-71.

Lewis, T.G. *Pascal Programming for the Apple*. Reston: Reston Publishing Co., 1980.

Lewis, T.G., and M.Z. Smith. *Applying Data Structures*. Boston: Houghton-Mifflin Co., 1982.

Martin, W.A., and D.N. Ness. "Optimizing Binary Trees Grown with a Sorting Algorithm," *Comm ACM 15*.

Maurer, N.D. "AVL Trees," *Byte* (June 1983):387-90.

Ottman, T., H.W. Six, and D. Wood. "Right Brother Trees," *Comm ACM* (September 1978).

Sussenguth, E.H. "Use of Tree Structures for Processing Files," *Comm ACM 6*. 272-79.

Atindra Chaturvedi is a database administrator for United States Lines. He has a bachelor's degree in chemical engineering and an MBA in management information systems.

PC-AT®

The ultimate in PC performance.

Introducing Advantage! From AST.

The first and only multifunction board for your PC-AT, Advantage! lets you add millions of characters of memory capacity, two serial ports, a parallel port and a game port. All in a single expansion slot.

The Memory Advantage! Advantage! provides the extra memory you need to run sophisticated integrated software such as Framework,™ Lotus 1-2-3™ and Symphony.™ To operate multitasking programs such as IBM's TopView™ or multiuser operating systems such as XENIX.™ To handle larger amounts of data, faster. Or for RAM disks.

Our unique memory addressing technique rounds out your system board to 640K, and continues memory expansion at 1 Megabyte. You can add up to 3 Megabytes of parity checked user memory in a single slot using combinations of 64K or advanced 256K memory chips. And of course, Advantage! supports your AT's high performance bus and faster program processing speed.

The I/O Advantage! Connect printers, plotters, mice and joysticks. Add a modem to tap into CompuServe,® EasyLink,© Dow Jones® and other information networks. Or with the appropriate software, connect other terminals and create multiuser environments.

An AT compatible serial port and a parallel port are standard Advantage! features. With our optional second serial port you can attach even more peripherals,

Advantage!™

Ultimate performance enhanced.

while our optional game port lets you plug in joysticks and other cursor-control devices for business or just for fun.

The Future Advantage! Advantage! from AST means your future is open to expansion. When you use precious slot space to your best advantage, you save space for the state-of-the-art in your PC-AT's future.

The Quality Advantage! AST's reputation is built on quality products, quality support and quality service. Our complete documentation makes the Advantage! exceptionally easy to install and use, but if it's not enough we're always here to help.

The Advantage!—only from AST. It's the advantage your IBM PC-AT needs today. Call our Customer Information Center (714) 863-1333 Ext. 5249 for more information on the Advantage! multifunction board and other AST products, or the location of your nearest dealer.

AST Research, Inc., 2121 Alton Avenue, Irvine, CA 92714 TWX: 753699AST UR

FEATURES

Memory Expansion
- 128Kb to 3.0Mb in a single slot
- User Upgradeable with either 64K or 256K memory chips
- Split Memory Addressing rounds out AT's system memory to 640K and continues memory expansion at 1Mb

I/O Expansion
- Up to 2 Serial Ports (1 optional)
- Parallel Printer Port
- Optional Game Port

Advantage! Supports AT's Full Program Processing Speed

AST RESEARCH INC.

Environmental Excavations

Accessing Command Processor's SET parameters with IBM Pascal

ROBERT B. STAM

The command processor (COMMAND.COM) of DOS versions 2.0 and later keeps an environment of parameters that can be updated using the SET command. The *DOS 2.0 Reference Manual* says the following, "In this way (using the SET command), it is possible to enter keywords and parameters that are not meaningful to DOS, but can be found and interpreted by applications that are designed to examine the environment."

What follows is an explanation of how Pascal programs can examine the environment; included is a subroutine that does so looking for a parameter of a given name. The subroutine is implemented as a Pascal unit. The advantage of using units is that once a unit has been created, it can be used in any number of programs.

SET is used to enter new parameters in the environment, as well as to modify or remove existing ones. When used with no arguments, it lists the current environment. The syntax is:

```
SET [name=[parameter]]
```

If **name** does not exist, the name and the parameter are added to the environment. If **name** does exist, the value of the parameter is changed. If **name** is used without a **parameter** value (but with the equal sign), both the name and the parameter are removed from the environment.

The command processor always converts **name** to uppercase, but uses **parameter** exactly as it is entered, so uppercase names must be used when examining the environment. As an example of how this command might be useful, consider programs that produce printed output. If there is more than one printer, the user would want a way to select which printer should be used.

A convenient way to do this is to have programs examine the environment for a name called PRINTER, and to use the parameter found to decide which printer to use. To make programs use the second printer, the following SET command could be used:

```
SET PRINTER=LPT2
```

In designing programs that examine the environment, a programmer should always allow for the absence of the name in the environment and should use an appropriate default. For example, if the name PRINTER was not found in the environment, printed output would probably be sent to LPT1.

The environment is not initially empty, although it can become so if all entries are removed. Normally, at least one entry, COMSPEC, will be there. It describes the path DOS uses to locate COMMAND.COM for reloading.

The PATH and PROMPT commands also use the environment, and store the search paths and the prompt string in the environment under the names PATH and PROMPT respectively. Programs can examine the environment for these names just like any others.

The command processor initially reserves 127 bytes for storing the environment, not very much if the environment is to be used a lot. When the environment gets full, the command processor attempts to expand it, but cannot always do so. In particular, if any program has terminated and stayed resident since DOS was started, it will no longer be possible to expand the environment. Many utility programs stay resident, as do several DOS commands, such as PRINT and GRAPHICS.

If the environment will be used a lot, it should be forced to grow before any program is executed that will terminate and stay resident. This can be done using a series of dummy SET statements such as

```
SET 1=xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
SET 2=xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

and so on. These names can be removed from the environment at a later time, thus freeing the space that they occupied for other names. Unfortunately for the user, dummy statements, such as the examples given above, cannot be used in a batch file, because the batch file processor temporarily blocks expansion of the environment.

The environment unit provides easy access to the environment via a single procedure called **getparameter**. All the user does is provide a name; **getparameter** will look in the environment for a parameter with that name.

Listing 1 is a sample program that shows how to examine the environment for the PRINTER parameter, including how a default value can be assigned if the parameter is not found. Any other parameter can be found using the same program by simply changing the name of the parameter.

Getparameter is a procedure, not a function. It returns the parameter value found in the environment by modifying it: the previous value of the parameter variable is lost. Both arguments to **getparameter** are LSTRINGs, the IBM Pascal date type for variable length strings. This is just what is needed for handling names of different sizes and parameter values of different lengths.

Getparameter is "bullet-proof" in the sense that it cannot fail in a drastic way. The only errors that can occur are (1) the name does not exist in the environment, or (2) the parameter found does not fit in the parameter variable provided. If the name provided does not exist, the parameter variable is set to the null string. Since this is the only time a null string will be returned, it is a direct indication that the name was not found in the environment. If the parameter variable provided is not large enough to hold the parameter found, then only as much of the parameter value as fits is returned. Always use a variable large enough to hold the largest parameter value expected.

USING UNITS

All units must have an interface description that describes the unit to programs wishing to use it. Listing 2 is the interface description file of the environment unit. All units must also have an implementation part that is compiled separately, producing an object module that must be linked with programs using the unit. Normally, the implementation is written in Pascal, but it may be written in assembly language when greater efficiency is necessary. Listing 3 is the implementation of the environment unit, written in Pascal.

If a program uses the environment unit the compiler must be informed by including the interface file at the beginning of the program and by placing a **uses** statement after the program header. For placement of these statements, see the sample program in listing 1.

When linking the program, the object module for the environment unit must be included. For the sample program in listing 1, this means using the following link command:

```
LINK GETPRINT+ENVIRON;
```

If the environment unit will be used a lot, it might be advisable to include the environment unit object module in the IBM Pascal Library so LINK can find it automatically. (See the *IBM Pascal 2.0 Fundamentals Manual* regarding the new library manager.) The following sequence of commands illustrates the use of the GETPRINT program:

```
C>SET PRINTER=LPT2

C>SET
COMSPEC=C:\COMMAND.COM
PATH=C:\COMMANDS;C:\
PRINTER=LPT2

C>GETPRINT
LPT2

C>SET PRINTER=
```

ILLUSTRATION • TOM CURRY

FIGURE 1: Use of SET Commands

'P	'R	'I	'N	'T	'E	'R	'=
'L	'P	'T	'2	00	'D	'I	'R
'E	'C	'T	'O	'R	'Y	'=	'C
':'	'\'	'L	'E	'D	'G	'E	'R
00	'F	'O	'R	'M	'A	'T	'=
'C	'O	'M	'P	'R	'E	'S	'S
'E	'D	00	00				

The SET commands, SET PRINTER=LPT2, SET DIRECTORY=C:\LEDGER, SET FORMAT=COMPRESSED, would result in an internal representation of the environment that looks like the above.

```
C>SET
COMSPEC=C:\COMMAND.COM
PATH=C:\COMMANDS;C:\

C>GETPRINT
LPT1

C>
```

The basic method for finding a parameter in the environment is to look at the strings one-by-one until a string with a matching name is found, or until the end of the environment is reached (indicating that the name is not in the environment). A couple of "dirty tricks" are needed in order for things to work correctly, having mostly to do with the differences between IBM Pascal STRINGs and LSTRINGs. In addition, some special IBM Pascal extensions are used to speed up the process. (Listing 3 is the Pascal code for the following implementation.)

The environment is stored in a block of memory reserved by DOS for this purpose. Each name and parameter is stored as an ASCII string that ends with a null byte; the environment itself ends with an additional null byte. Figure 1 illustrates an example of the use of SET commands. The null byte at the end of each string can be used to find the beginning of the next string; the final null byte signals the end of the environment itself.

Since this case will use a pointer to the environment, and since pointers must point to a specific data types, a data type for the environment must be selected. In this case, the environment was declared as a STRING, because individual characters can be selected by simple indexing.

Strings must have a fixed length declared, but the length of the environment is difficult to determine. The DOS manual says the environment will never be more than 32KB long. To play it safe, the environment is declared as a string of 32,767 characters, even though it usually will be much smaller. The user can always tell where the end of the environment is by the extra null byte. So even though the environment is declared as a very long string, it is the user's responsibility to limit access to the currently allocated space.

The segment address of the environment can be found at offset 2CH of the program prefix area, according to the *DOS Reference Manual.* Actually, the program prefix area points to a copy of the environment. When a program finishes execution, this read-only copy is thrown away; it is the original environment that is changed by future SET commands. Wherever it is, the environment is outside of the Pascal data space: normal pointer types cannot be used to point to it. An IBM Pascal extended pointer type (the ADS pointer) that can be used to access any location in memory will be discussed later.

The program prefix area can be used to find the segment address of the environment, but where is the program prefix area itself? IBM Pascal does not provide any help in finding the program prefix area; the user must look for it. For that matter, how does any program know where the program prefix area is? According to the *DOS Reference Manual*, the segment address of the program prefix area is in the DS and ES registers when a program begins execution.

It is not documented in the reference materials, but an examination of the IBM Pascal run-time initialization code (see ENTX6S.ASM that comes with IBM Pascal) shows that the initial value of the ES register is saved in a global variable called CESXQQ. Therefore, CESXQQ can be used to find the program prefix area, and, indirectly, the environment. The program prefix area has some other interesting information as well. See Tech Notebook #30 ("Retrieving Command Line Arguments from IBM Pascal 2.0, page 000) and the DOS documentation for more details.

ADS POINTERS

IBM Pascal has several extensions designed to make it more useful as a systems programming language, including two new pointer types that represent machine addresses. Regular pointer types can point only to data allocated in the Pascal heap, not to data located in other areas of memory.

The ADR pointer type can point to any address in the IBM Pascal data segment, including the heap, the stack, all static variables, and all constants. While this is a useful relaxation of the tight rules governing regular pointer use, it is not enough to allow pointing to the environment (because it is outside the IBM Pascal data segment).

The ADS pointer type can point to any memory address: these pointers are segmented machine addresses. The 8088 uses segmented addressing, and memory locations are addressed using a segment address and an offset. The segment address is sometimes called the paragraph address. Paragraphs are 16-byte blocks of memory at 16-byte boundaries, and the segment address is the paragraph number of the first paragraph in the segment. Segments normally begin on paragraph boundaries, and can be any size up to 64KB (in multiples of 16 bytes). The offset selects a particular byte within a segment, and is computed from the beginning of the segment.

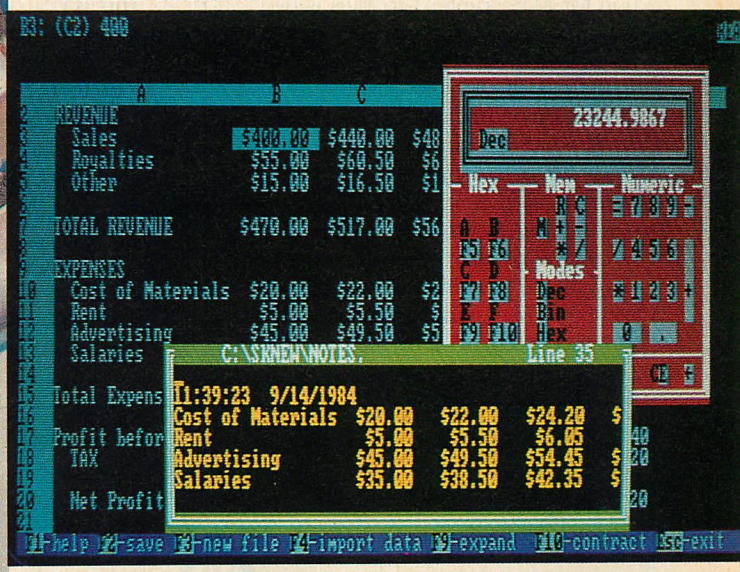
The physical memory of the PC uses linear addressing, so segmented addresses must be mapped to linear addresses before physical memory locations can be referenced. The 8088 converts segmented addresses to linear addresses by attaching four low-order zero bits to the segment address (in effect, multiplying it by 16) and adding the offset to this intermediate result.

$54.95
copy-protected

$84.95
noncopy-protected

30 DAY MONEY-BACK GUARANTEE

Here's Sidekick in action. That's Lotus 1-2-3 running underneath. In the Sidekick Notepad you can see data that's been imported from the Lotus screen. On the upper right, that's the Sidekick Calculator.



NOW . . . Whether you're working in dBase, Lotus, Wordstar or whatever . . . you can unleash the full power of your computer . . . and make a lot of extra space on your desk at the same time.

Whenever you're using your computer . . . from start to finish of your session . . . Sidekick™ will be there . . . ready to serve. And it's as lightning-fast and compact as only Borland knows how to make it.

There's a notepad that has a full-screen editor that can time and date stamp your notes, and then save them to disk. You can even pull information into the notepad directly from the screen of your "underlying" software.

Suppose you're working in Lotus and the phone suddenly rings. Give your Sidekick a call and it pops right up over Lotus with the notepad you need. Or an appointment calendar . . . one you can never misplace.

What if you need to do a quick calculation? A keystroke instantly brings up the calculator. And the results of your calculations can even be transferred to your "underlying" software.

Need to make a phone call? Up pops your personal phone directory. Type in the name you want . . . and Sidekick jumps right to the phone number. Another keystroke, and the phone is automatically dialed for you.†

There's lots more, too. You can move the Sidekick windows anywhere on the screen you like. And you can have as many on screen at a time as you need. There's even an on-line help window for each of Sidekick's features.

We designed it because we needed it. If you've ever been writing a report and needed to do a quick calculation, or jot down a note, then you need Sidekick, too.

†Only with Hayes Smartmodem and compatibles.

GOT *YOUR* SIDEKICK™ YET?

SIDEKICK IS AVAILABLE AT YOUR NEAREST SOFTWARE DEALER, OR DIRECT FROM BORLAND.

For VISA and MasterCard orders, call **1(800) 255-8008**: in California call **(800) 742-1133.**
Lines are open 24 hours a day, 7 days a week.

_____$54.95 SideKick copy-protected
_____$84.95 SideKick noncopy-protected
(Prices include $5 shipping and handling per order.)

Check ☐ Money Order ☐
VISA ☐ MasterCard ☐
Card # _____
Expiration Date _____

Please be sure your system is an IBM PC, AT, XT jr. or a true PC-compatible. PCjr. users must order noncopy-protected version.

NAME _____
ADDRESS _____
CITY/STATE/ZIP _____
TELEPHONE _____

California residents add 6% sales tax. Outside USA add $10; payment must be by bank draft payable in the US and in US dollars. COD and purchase orders will not be accepted.

CIRCLE NO. 108 ON READER SERVICE CARD

BORLAND INTERNATIONAL
Borland International
4113 Scotts Valley Drive
Scotts Valley, California 95066
TELEX: 172373

The IBM Pascal internal representation for ADS pointer variables is two consecutive 16-bit words. The first word is the offset; the second is the segment address. ADS pointer variables are also record structures with two fields of type **word**. The segment address is in the .S field; the offset is in the .R field. The value of an ADS pointer variable is normally changed using an assignment statement. In cases where it is not possible to use a regular assignment statement, the ADS pointer variable's value can also be changed by setting the segment and offset components directly.

This is perhaps the most confusing part of the implementation of the environment unit because it involves pointers to pointers, and the levels of indirection can get confusing. First, the declarations of the variables used are very important. The declarations are

```
CESXQQ [extern]    : word;
environmentaddress : ads of word;
environment        : ads of string (32767);
```

CESXQQ contains the segment address of the program prefix area, the **environmentaddress** variable points to the word of the program prefix area that has the segment address of the environment, and the **environment** variable points to the environment itself.

The code used to initialize these pointers is the following:

```
environmentaddress.S := CESXQQ;
environmentaddress.R := #2C;
environment.S := environmentaddress^;
environment.R := 0;
```

The first two assignment statements initialize the pointer to the word of the program prefix area that has the segment address of the environment. This is accomplished by setting the segment component to the segment address of the program prefix area, and the offset component to the desired offset.

The last two assignment statements initialize the pointer to the environment itself by using the first pointer to read the segment address of the environment from the word at offset 2CH of the program prefix area. The offset component of the pointer to the environment is simply set to zero.

The **environment** pointer is now pointing to the first byte of the environment; it remains constant for the remainder of the implementation. Pointing within the environment is accomplished using normal string indexing. If necessary, this also could be done by changing the offset component of the **environment** pointer. Because the **environment** pointer is never changed, its value is computed once in the initialization section of the implementation and remains constant for the duration of program execution.

The method used to examine the environment depends on looking at each string individually; a convenient way is needed to point at the current string, and to move on to the next one. The variable I points to the first character of the current string. Because the environment is a string, the I is also used as an index into the environment. The variable L has the size of the current string; this is used to ignore any strings that are too short. A string must be at least as long as the name that is being searched or it can be skipped. To move on to the next string, simply add L+1 to I. The extra 1 steps over the null byte at the end of the string.

The IBM Pascal extension **scaneq** is useful to find the end of the current string. **Scaneq** scans a string for a character equal to the search character, and returns as its value the number of characters skipped over. Normally **scaneq** must be told the maximum number of characters it can scan over. In this case,

the maximum is not known, so it is told to scan over 32,767 characters. Any number at least as large as the environment would do, so, to be safe, the largest possible integer is used.

Scaneq can be told to scan over so many characters because it is assumed that the environment is well-formed, and that there is a null byte at the end of the current string. If for some reason the environment is not well-formed, **scaneq** would go on searching beyond the end of the memory block allocated to the environment. Because a null byte is being searched for, the search character is expressed as chr(0). The internal representation of strings is such that each character is stored in one byte, so searching for a null character is the same as searching for a null byte.

The string being searched is the environment; it is specified by dereferencing the **environment** pointer. **Scaneq** wants the address of the beginning of the string, even if the user wants to start scanning from somewhere in the middle of the string. The variable I points to the current string and is where this scan will start, so it is the final argument to **scaneq**. **Scaneq** returns its value (the length of the current string) which is assigned to variable L.

In the case of an empty environment, **scaneq** would skip no characters before finding the null byte at the end of the environment, and would return a value of zero—a direct indication to the user that the environment is empty.

LSTRING SPLICING

Normally, regular assignment statements are used to change the value of an **lstring** variable (or some IBM Pascal extension such as **concat**, **delete**, or **insert**). In this case, the **lstring** variable that needs to be modified is the parameter variable. But, the regular assignment statements cannot be used for the parameter variable because it is a super array (in that the maximum length of the **lstring** is not declared), and IBM Pascal does not allow super arrays on either side of an assignment statement.

The probable reason for this is that the compiler cannot know the maximum length of the parameter variable that will eventually be provided and, therefore, cannot generate safe code for the assignment statement. To circumvent this technicality, assignment to the parameter variable can be simulated by "splicing" the internal representation of the **lstring**. Some IBM Pascal extensions can directly modify the memory contents of the parameter variable.

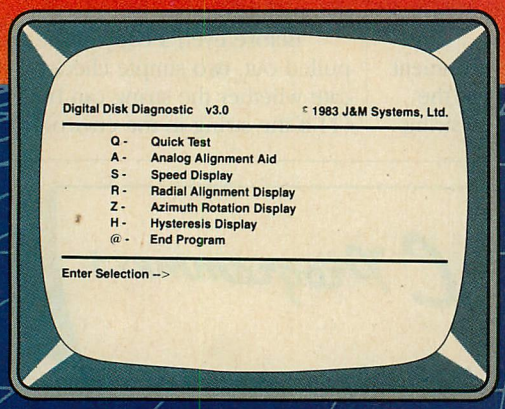
The current length of an **lstring** can be changed by modifying its first byte. The value the user stores in this byte becomes the new current length of the **lstring**. Care must be taken that the new length is not longer than the maximum length allowed. The value of an **lstring** can be changed by directly modifying the memory contents of the remaining bytes. This could be done one character at a time using string indexing, but instead we will use an IBM Pascal extension called **moves1**.

Before a string in the environment can be searched for a parameter, the name must be pulled out of the string. IBM Pascal provides no way to directly compare an **lstring** (the name being searched) with a substring of a **string** (the name in the environment string). The solution is to pull the name out of the environment and put it in an **lstring**, so the two **lstrings** can be compared. The variable copy is used to hold a copy of the name pulled out of the environment.

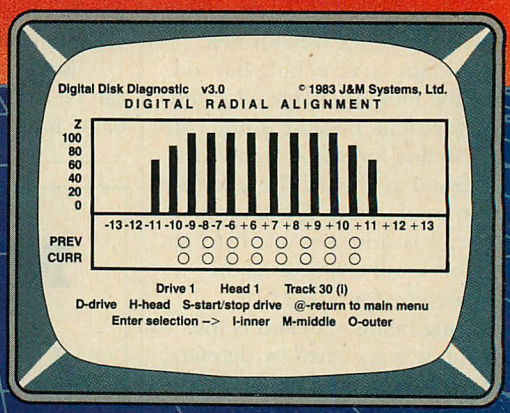
Before even a copy of the name is pulled out, two simple checks will indicate whether the string can be ignored: (1) if the string in the environment is

DISK DRIVE ANALYSIS PROGRAM

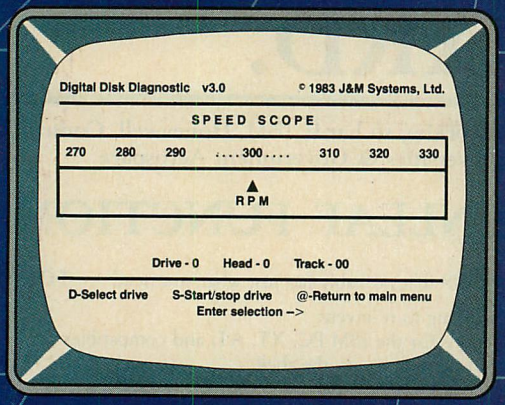
. . . A UNIQUE APPROACH TO DISK RELIABILITY!



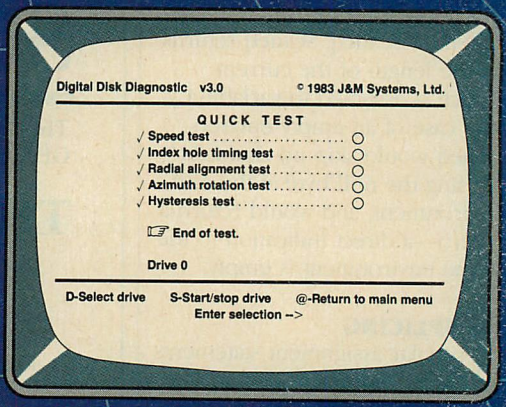
Select any one of seven tests to perform preventive maintenance or to isolate problems. Simple, single-letter commands make DDA easy to use! Use DDA to align the head, or adjust the speed.



Use the DDA Radial Alignment Test to check the head alignment of your drives. No need for an oscilloscope or other expensive test equipment!



Check the motor speed of your drives. Or, you can even use the Speed Test to adjust the drive speed. No need for any test equipment!



Use the Quick Test to quickly and automatically test five of the most important performance parameters of your drive. Monitor your drives for long term drift. Isolate problems quickly and automatically!

PROTECT YOUR DATA.

Now you can make sure your data is being recorded properly by the use of the revolutionary ***Disk Drive Analysis Program.***

The Disk Drive Analysis Program from J & M Systems, tests your disk's performance and calibration without any additional equipment! It measures your disk's performance and displays it on your screen.

This is the most comprehensive disk diagnostic program available for your IBM microcomputer. You can even adjust drive alignment while watching the display!

Spot problems ***before*** they endanger your data!

If you own a disk drive, you ***need*** the ***Disk Drive Analysis Program!***

IBM	Price #
1 - 160K Drives	$79
2 - 320K Drives (Also test 160K drives)	$99

DDA also available for other models

J & M SYSTEMS IS THE DRIVING FORCE!

J & M SYSTEMS, LTD.

137 UTAH NE • ALBUQUERQUE, N.M. 87108 • 505/265-1501

CIRCLE NO. 205 ON READER SERVICE CARD

shorter than the name being searched, or (2) if the name in the current string is not the same size as the name being searched. Variable N holds the size of the name being searched; the easiest way to see if the name in the current string is the right size is to look N characters beyond the beginning of the current string for an equal sign.

Once the name is found, variable P holds the size of the parameter. This is easy to figure out because the size of the current string is already known. Subtracting the size of the name from the size of the string, plus one more for the equal sign, gives the size of the parameter. Before pulling out the parameter, variable P must be compared to the maximum allowed size of the parameter variable. Never pull out more characters from the environment than will actually fit in the parameter variable.

String indexing could be used to pull out a name or parameter one character at a time, but it is much faster to use **moves1**, which moves around arbitrary areas of memory. Care must be taken in moving memory in this way; if it lands in the wrong place a lot of damage can be done. To pull the name out of the current string, the source address is computed by dereferencing the **environment** pointer and subindexing it to the first character of the current string. The destination address is the first character of the variable **copy**. The number of characters to move is given by the variable N.

*String indexing could be used to pull out a name or parameter one character at a time, but it is much faster, albeit more dangerous, to use **moves1**, which moves arbitrary areas of memory.*

To pull the parameter out of the current string, the source address is computed by dereferencing the **environment** pointer and subindexing it to the first character of the parameter. The first character of the parameter is N+1 characters beyond the first character of the current string. The destination address is the first character of the parameter variable. The number of characters to move is given by the P, which has already been checked against the maximum size of the parameter variable.

If the parameter is found, it is copied from the environment to the parameter variable, where it can then be easily used. In the event that the parameter is not found, then "splicing" is used to set the parameter variable to the null string, which is accomplished easily by setting the first byte to zero.

The command processor of DOS keeps an environment of parameters that is easily modified using the SET command; using SET, parameters can be passed to applications that are designed to examine the environment, and execute the program accordingly.

The environment is also easily examined from Pascal programs using the environment unit; units help to simplify complex tasks. It is not necessary to know how this unit does its "dirty work" (and it does get dirty) in order to use **getparameter**, the procedure that makes it easy for any Pascal program to get parameters from the environment. So, Pascal users, take advantage!

Robert B. Stam is a graduate of Harvard College and is vice-president of Futureware, Inc. of Jackson, Mississippi. He has taught several computer science courses at the University of Costa Rica in his home country.

LISTING 1: GETPRINT.PAS

```
(*$INCLUDE:'environ.int'*)

program getprint (input, output);
uses environment;

var printer:lstring(127);

begin
  getparameter ('PRINTER', printer);
  if (printer = null) then printer := 'LPT1';
  writeln (printer);
end.
```

LISTING 2: ENVIRON.INT

```
interface;
unit environment (getparameter);

procedure getparameter (const name   :lstring;
                        var parameter:lstring);

begin
end;
```

LISTING 3: ENVIRON.PAS

```
(*$INCLUDE:'environ.int'*)

implementation of environment;

var

CESXQQ [extern]    : word;
environmentaddress : ads of word;
environment        : ads of string(32767);

procedure getparameter;

var I,L,N,P,M:integer;
var copy:lstring(255);
var found:boolean;

begin
  I := 1;
  N := ord(name[0]);
  M := upper(parameter);
  found := false;

  repeat
    L := scaneq(32767,chr(0),environment^,I);
    if (L > N) and (environment^[I+N] = '=') then
      begin
        copy[0] := chr(N);
        movesl (ads environment^[I],
                ads copy        [1], wrd(N));
        if (name = copy) then
          begin
            P := L-N-1; if (P > M) then P := M;
            parameter[0] := chr(P);
            movesl (ads environment^[I+N+1],
                    ads parameter   [1    ], wrd(P));
            found := true;
          end;
      end;

    if (L > 0) then I := I+L+1;
  until found or (environment^[I] = chr(0));

  if not found then parameter[0] := chr(0);
end;

begin
  environmentaddress.S := CESXQQ;
  environmentaddress.R := #2C;
  environment.S := environmentaddress^;
  environment.R := 0;
end.
```

Make room for a new assistant.

Introducing the IBM PC Engineering/Scientific Series.

It puts answers on your desktop.

The IBM Personal Computer Engineering/Scientific Series is an unusually flexible and talented assistant.

As you can see from the box on the right, you can select exactly the workstation you need from a set of compatible engineering and scientific enhancements to the IBM Personal Computer Family. Configure an IBM Personal Computer to help with product design, perform numerical analysis, or monitor and control laboratory experiments.

For the reasonable cost of IBM PC/ES Series enhancements to an IBM Personal Computer, you can do jobs that traditionally demanded time on a mainframe. Display data and plot results as they happen. Control both the business and technical sides of your job from one compact workstation.

And if you want host or networking communications, the IBM Personal Computer has the right connections.

Advanced color graphics.

The IBM PC/ES Series is designed to make a graphic impression in jobs ranging from computer aided design applications and image processing to business presentations.

The IBM PC Professional Graphics Display can make even good results look better with a 640 x 480 screen and up to 256 simultaneous colors available from a palette of 4,096. An advanced anti-glare screen makes the Professional Graphics Display easy to work with and easy on your eyes.

The IBM PC Professional Graphics Controller's onboard 8088 processor can help improve productivity by speeding throughput of even the most sophisticated graphics projects. The graphics firmware support, resident on the adapter, includes 2D and 3D graphics primitives, programmable display lists, and a definable character set to help make your graphics work more creative.

And there's an emulation mode that gives you immediate access to the entire range of graphics programs already written for the IBM Personal Computer.

Productive programming environment.

IBM PC/ES Series layered graphics software is geared for increased programming productivity.

It supports a 32K x 32K addressable point data base, and includes high-level graphics subroutines that are consistent with the Draft ISO and ANSI GKS Standards.

The IBM PC Graphics Development Toolkit provides a device-independent programming environment to help simplify graphics programming tasks. Device independence also helps promote program longevity, since your graphics applications can be programmed to interface with future generations of graphics devices.

IBM Personal Computer AT with Engineering/Scientific Series components.

Precise high-speed computation.

IBM PC Professional FORTRAN by Ryan-McFarland Corporation—a full ANSI 77 implementation plus enhancements—gives your applications optimized performance on the IBM Personal Computer.

Professional FORTRAN helps you get precise results when you want them, without unnecessary use of valuable mainframe time.

It enables you to begin work immediately on large or small host programs or to recompile existing FORTRAN programs to run on an IBM PC with Engineering/Scientific enhancements. And Professional FORTRAN allows you to utilize all the IBM PC/ES Series graphics development tools in your desktop application work.

Accurate instrumentation.

The IBM PC/ES Series gives you a single point of control for a laboratory full of work.

You can control and monitor laboratory or industrial processes, automated electronic testing, and the status of ongoing experiments via the IBM PC Data Acquisition and Control Adapter. Programming support provides flexible high-level and low-level language interfaces. Powerful built-in diagnostics make installation of the Data Acquisition and Control Adapter quick and easy.

A single half-size IBM PC General Purpose Interface Bus Adapter lets you interactively control instruments as diverse as plotters and spectrum analyzers through application programming. High-level and low-level language access makes it easy to plot and display data. Menu-driven supporting software includes a versatile hardware configurator to aid in installation.

IBM Personal Computer Engineering/Scientific Series Components

Processors
IBM Personal Computer AT
IBM Personal Computer XT
IBM Personal Computer with expansion unit

Color Graphics Hardware
IBM Personal Computer Professional Graphics Display
IBM Personal Computer Professional Graphics Controller

Color Graphics Software
IBM Personal Computer Graphical Kernel System
IBM Personal Computer Graphical File System
IBM Personal Computer Plotting System
IBM Personal Computer Graphics Terminal Emulator
IBM Personal Computer Graphics Development Toolkit

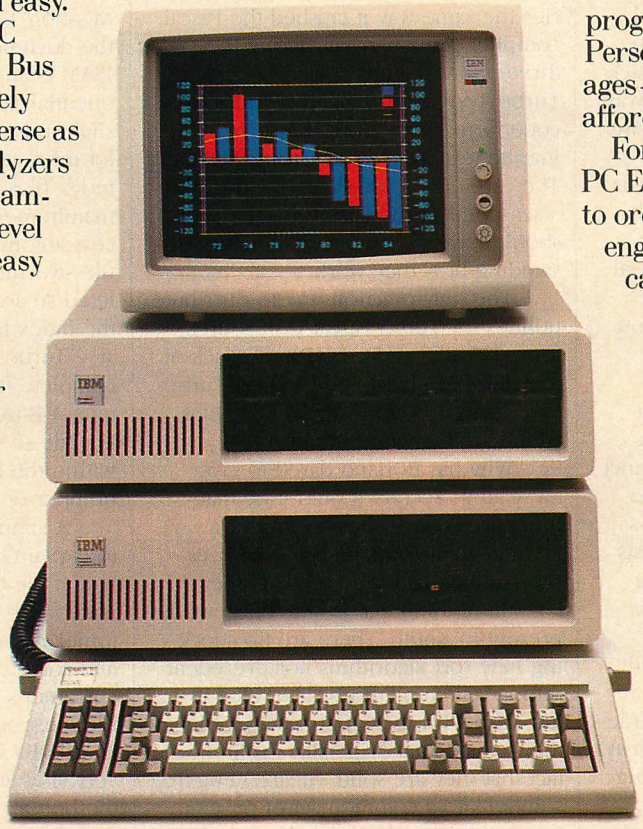
High-Speed Computation
IBM Personal Computer Professional FORTRAN

Instrumentation/Data Acquisition
IBM Personal Computer Data Acquisition and Control Adapter
IBM Personal Computer Data Acquisition and Control Adapter Distribution Panel
IBM Personal Computer Data Acquisition and Control Adapter Programming Support
IBM Personal Computer General Purpose Interface Bus Adapter
IBM Personal Computer General Purpose Interface Bus Adapter Programming Support

Complete the equation.

This assistant is ready to start work immediately. There are over 350 independently developed engineering and scientific application programs—plus many IBM Personal Computer business packages—available today for the affordable IBM PC/ES Series.

For information about the IBM PC Engineering/Scientific Series or to order a catalog of available engineering/scientific programs, call 1-800-447-4700. Or write: IBM Corporation, Dept. ES, 1909 East Cornell, Peoria, Illinois 61614.



IBM Personal Computer with Expansion Unit and Engineering/Scientific Series components.

IBM®

CIRCLE NO. 138 ON READER SERVICE CARD

Tools for the Pascal Programmer

Borland International's newest product, Turbo Toolbox, may not be the breakthrough that Turbo Pascal was, but it is solid and useful.

JEFF DUNTEMANN

If there is any way to characterize the products coming out of Borland International right now, it would have to be a spirit of get-the-job-done pragmatism. Turbo Pascal and Sidekick have done more for my productivity as a programmer than any other software products I have ever used. Borland's newest product, the Turbo Toolbox, is not likely to have the impact of Turbo and Sidekick. It is a much more narrowly focused product and will appeal to a smaller audience. On the other hand, if you find that you need it, you will have a great need for it.

As its name implies, the Turbo Toolbox is a collection of utility routines for use with Turbo Pascal. The routines fall together into three distinct utility products: Turbo Access, Turbo Sort, and the General Installation System. The distribution package consists of a single diskette and a 96-page paperback book. The Turbo Toolbox is the same price as Turbo Pascal and Sidekick: $49.95. It must be used with Turbo Pascal release 2.0.

Unlike either Turbo Pascal or Sidekick, the Toolbox requires a good deal of study and some basic programming knowhow before it will pay off. I worked intensely with the Toolbox for two weeks and got most of the way up the learning curve. An applications programmer who spends the time to know the Toolbox well will most likely find it more than worth the effort.

I sincerely hope that Borland crashes the source-code paranoia barrier the same way it crashed the Pascal compiler price barrier. Two of the three utility systems that make up the Turbo Toolbox are given in full source code form, ready to read just by TYPEing the text files to the screen or paper. They are only sparsely commented, but a seasoned Turbo Pascal programmer should have no trouble at all understanding how they work.

The only practical way to distribute utilities for Turbo Pascal is in source form, because of Turbo Pascal's lack of any linkable object code format. Turbo Pascal can produce only ".COM" files, which are by nature unlinkable.

Why has Borland dared to give away all its secrets? Could it be that there aren't any "secrets" to give away? Having spent some time poring over the Toolbox listings, I realized they contain nothing that is not conceptually present in Knuth. There are no shocking new sort algorithms nor precedent-setting search methods. The value in the Turbo Toolbox routines is not in their conceptual originality but in the fact that they are solid, (nearly) bug-free solutions to a common set of programming problems. Borland stands to lose nothing by publishing its source code.

TURBO ACCESS

The largest and most complicated of the three components of the Turbo Toolbox, by a wide margin, is Turbo Access. The Turbo Toolbox ads and literature mention Turbo ISAM, but nothing in the documentation explains what it is. ISAM is an acronym for Indexed Sequential Access Method, and that is a slight misnomer—Turbo Access does a lot more automatic maintenance of the index files than the well-known IBM mainframe ISAM systems do. Turbo Access automatically reorganizes the index file structure as keys are added and deleted so access times to a given item in the index file are as small as possible.

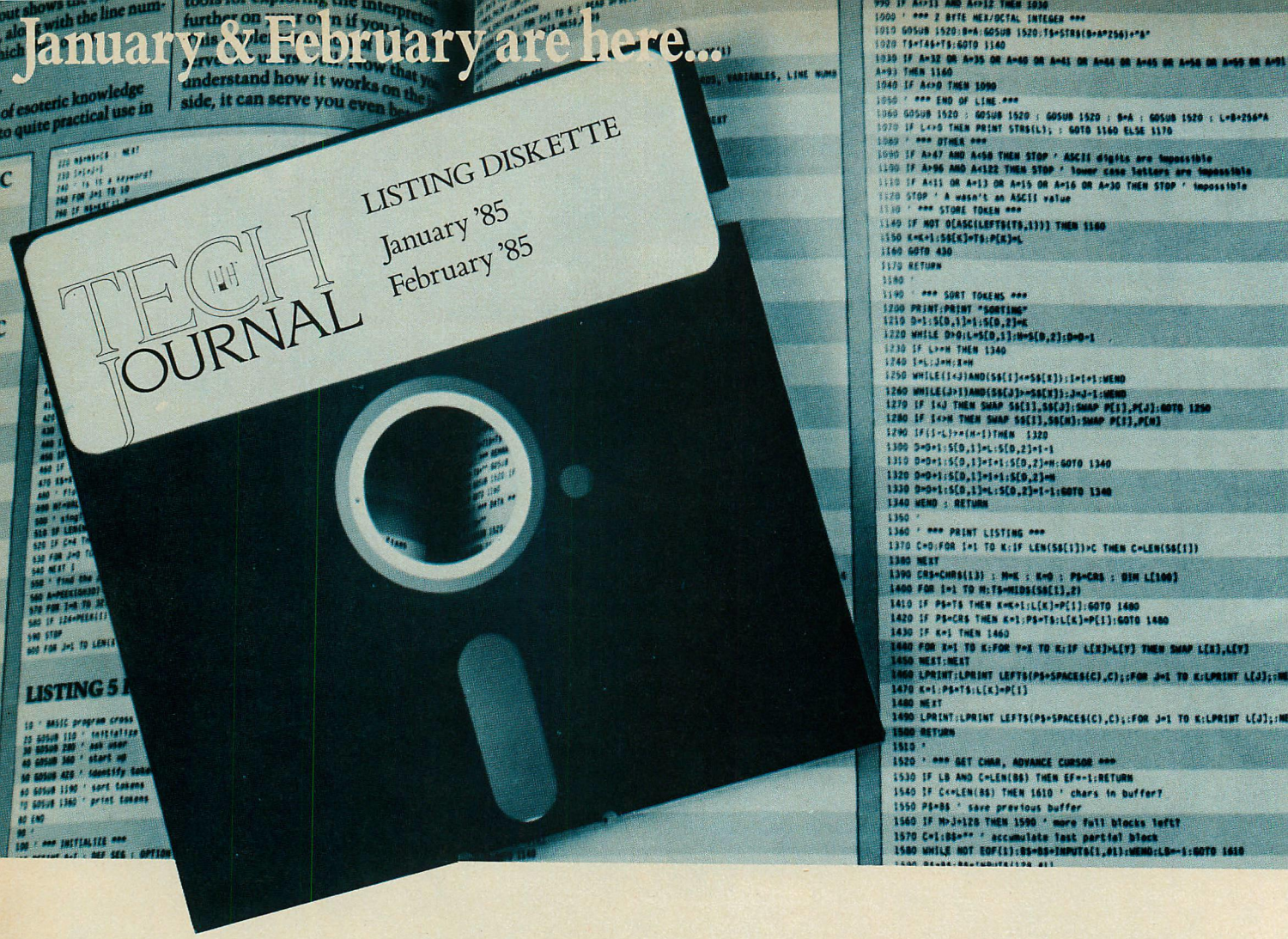
Turbo Access actually provides a complete B-tree file management system. A B-tree is a way of arranging a key file so that locating a given key within the file is done with the smallest number of disk accesses.

A simple example is the best way to explain the concept of B-trees. Start with a file of data records, in no particular order. Each record contains a customer name, address, phone number, and some other information. To find a customer's telephone number, the data file must be read record by record. This is called a *sequential search* and can be very slow if the needed record is near the bottom of a large file.

One way to speed up the process is to extract all the names from the data file into a separate file, with a record number next to each name in the new file. The names are called keys; a file of such records is a key file. The record

ILLUSTRATION • TOM CURRY


CURRY



PC Tech Journal Program Listings on Diskette.

CALL TOLL-FREE 1-800-526-0790

New Jersey residents: (201-540-0445)

OR MAIL COUPON BELOW

THAT'S RIGHT! You don't have to type in detailed program listings from this February issue. You can order them on a LISTING DISKETTE™ from PC Tech Journal. And on the same diskette you'll also receive the program listings for last month's issue.

Certain listings require use of other material (e.g., Pascal compiler). Please consult the relevant article in the magazine.

For your added convenience, we also offer two other LISTING DISKETTES from previous issues. They're all listed in the coupon. To order, simply fill it out. Charge-card customers can also use our toll-free telephone numbers. (Sorry, no C.O.D. or "bill me" orders.)

TECH JOURNAL

PC TECH JOURNAL
Listing Diskette Service
39 East Hanover Avenue
Morris Plains, NJ 07950

Please send me the following Listing Diskettes:

- ☐ No. 7—Jan. & Feb. '85 issues. $19.95
- ☐ No. 6—Oct., Nov. & Dec. '84 issues . . . $19.95
- ☐ No. 5—Aug. & Sept. '84 issues. $19.95

Total $_____

☐ Check Enclosed ☐ Please charge my credit card:
☐ MasterCard ☐ Visa ☐ American Express

No._____ Expiration Date_____

Signature_____

Name_____

Company_____

Address_____

City_____ State_____ Zip_____

Please allow approximately four to six weeks for delivery.

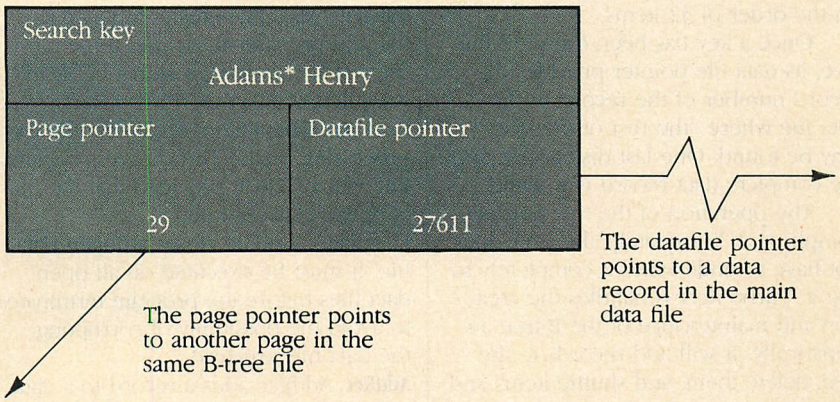
number next to each key in the key file allows the user to go to its associated record in the data file. The record number is known as a *pointer* from the key record into the data file.

A sequential search on the key file is faster than one on the data file because the key file records are smaller. Still, in a large file, the user may read past thousands of key records before finding the right one.

Comparing two key values in RAM is very fast compared to loading key values into RAM from disk. To speed up the search, the number of times needed to go out to the disk to load more keys into RAM must be minimized. To do this involves structuring the key file into what is known as a *tree*.

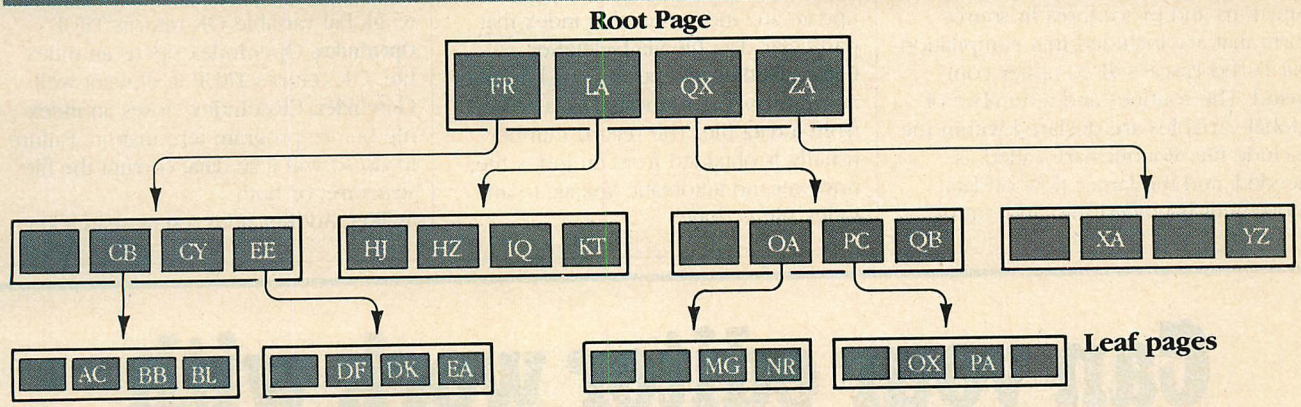
A tree is made up of items, each of which is a separate record in the key file (see figure 1). The item record contains the search key and two pointers. The *data pointer*, points to the record in the data file that contains the rest of the information associated with the search key. The page *pointer* points to another item in the key file.

FIGURE 1: *An Item*



Each item, composed of a search key and two pointers, is a separate record in a key file.

FIGURE 2: *A B-Tree*



The tree is searched for record NR by comparing it against the search key in each page, starting with the smallest.

With all the pointers correctly set up the numerous items in the key file can be spread out into a diagram such as the one shown in figure 2.

The oblong boxes are called *pages*, each containing a certain maximum number of items. This number varies from tree to tree, but it is always the same within a given tree, and it must always be an even number. The number for the tree in figure 2 is four. The Turbo Toolbox allows up to 254 items per page, although the optimum number usually falls between 16 and 32.

Physically, a page is a sequence of key file records. An entire page is always read from disk as a unit. The items in a page are in sort order. A page need not be completely full; shaded items in the B-tree in figure 2 are *holes* in the tree where items may be inserted in the future.

How does the tree work? To find out if the search key NR is present in the tree, begin by reading the root page from disk into memory. Compare NR against the search keys in the page, starting with the smallest. FR and LA are read and discarded, as they are less than NR. QX, however, is larger.

Since the keys in each page are in sort order, the user now knows that NR is not in the root page. Somehow he will have to start searching the leaf pages. The tree is set up in such a way that a leaf page always contains search keys *less* than the item the leaf page branched from. This means it is not necessary to search the leaf page pointed to by LA, because all of its search keys will be less than LA, and NR is greater than LA.

NR is less than QX, so if NR is in the file, it is on a path downward from QX. Thus, the leaf page pointed to by QX should be read. This time, after the first comparison, the user knows NR is not in that leaf page: NR is less than OA. So the leaf page pointed to by OA should be read from disk. On the second comparison, NR is found. Only three disk accesses and six comparisons were needed to pinpoint one search key out of 26 in the whole tree.

As the number of items per page increases, the number of disk accesses required to find a key goes down, and the number of comparisons goes up. But because disk accesses are *enormously* slower than comparisons, increasing the number of items per page makes for faster searches.

Taken to an extreme, the entire key file would exist in memory as a single page, and (assuming floppy disks) it might *still* be possible to search the en-

tire file sequentially in less time than it would take to perform a second disk access. In practice, a compromise value for Turbo Access pages is somewhere on the order of 32 items.

Once a key has been found in the tree, its data file pointer provides the record number of the record in the data file where "the rest of the story" may be found. One last disk access and the complete data record is in hand.

The operation of the tree is fairly complicated. Fortunately, the user does not have to understand it completely to use it. Turbo Access handles the creation and maintenance of the B-tree automatically. It will add records to the tree, delete them, and shuffle items and pages around so that the tree is always roughly balanced from right to left—the optimum shape for locating records in the least possible time.

PROGRAMMER INTERFACE

Turbo Access is presented as a set of functions and procedures in source form that are included in a compilation via Turbo Pascal's $I compiler command. The routines and a number of global variables are declared within the include file. Routines are called as needed, and the larger flow of data movement is guided into and out of data and key files, allowing Turbo Access to handle the details.

The individual routines in Turbo Access are as follows:

MakeFile. MakeFile creates a new data file and prepares it for use. If the file is created successfully, a global boolean variable OK is set to TRUE.

OpenFile. OpenFile opens a data file for processing by the Turbo Access system. OK returns TRUE provided that the file is opened successfully.

CloseFile. CloseFile closes an open data file. It must be executed on all open data files before the program terminates to avoid the possibility of corrupting the data file structure.

AddRec. AddRec adds a record to a data file and returns a record number, which must be passed to any index files that index that particular data file. This is done with the AddKey procedure.

DeleteRec. DeleteRec removes a record from a data file. Like AddRec, it returns a record number that must be used to update any index files that index that particular data file via DeleteKey.

GetRec. GetRec accepts a record number and returns the requested data record from a data file. The record number usually is obtained from an index file; however, no automatic linkage to an index file is done.

PutRec. PutRec writes a data record to a particular record number in the data file. This is an "update in place" routine for a record that is known to exist at a particular record number, usually as the result of a search of an index file. To add a completely new record to the data file AddRec must be used.

FileLen. FileLen returns the number of records allocated to the specified data file. This figure includes the reserved record 0 (used by Turbo Access for housekeeping information) and any deleted records not yet re-used.

UsedRecs. In contrast to FileLen, UsedRecs returns the number of records actually containing valid data. This figure excludes reserved and deleted records.

InitIndex. A number of tables and global variables are necessary to Turbo Access's index file management scheme. InitIndex sets them up prior to the opening of any index file.

MakeIndex. MakeIndex creates a new index file. If this is done without difficulty, global variable OK returns TRUE.

OpenIndex. OpenIndex opens an index file. OK returns TRUE if all went well.

CloseIndex. CloseIndex closes an index file before program termination. Failure to do so will lose data, corrupt the file structure, or both.

AddKey. AddKey adds a new search key

to an index file. OK will return FALSE if the key is a duplicate unless that particular index file was set up to allow duplicate keys. (This is done with MakeIndex or OpenIndex.)

DeleteKey. DeleteKey removes a key from an index file. If the index file allows duplicate keys, a record number must also be passed to DeleteKey, or an error will be generated and the key will not be removed.

FindKey. FindKey searches a key file and returns the record number of a search key that matches the given key. If no matching key is found. OK returns FALSE. If the index file has duplicate keys, only the first match encountered will be returned, because FindKey does not check for possible duplicate keys.

SearchKey. SearchKey will search an index file and return the first key found with a value greater than or equal to the given key. This allows you to locate records with only a partial key to work with. OK returns TRUE in all cases unless the given key is larger than every single key in the index file.

NextKey. NextKey must be used after either FindKey, SearchKey, or ClearKey. (These three routines set up an internal index file pointer that is needed by NextKey and PrevKey.) It returns the key following the last key fetched from the index file by whatever method. If NextKey is called with the internal pointer at the last record in the index file, OK returns FALSE and the pointer is back in its "limbo" position. Another call to NextKey at that point will wrap the pointer around to the first key in the file. This feature (along with a similar provision in PrevKey) allows an index file to be treated as a ring structure for sequential processing.

A record may be up to 65,535 bytes long. This is a constraint of the 8088's register system.

PrevKey. Just as with NextKey, PrevKey must have been preceded by at least one call to either FindKey, SearchKey, or ClearKey. It returns the key previous to the last key fetched by whatever method. If PrevKey is called with the internal pointer at the first record in the index file, OK returns FALSE and the pointer is back in its "limbo" position. Yet another call to PrevKey will wrap the pointer around and return the last key in the index file.

ClearKey. ClearKey prepares an index file for sequential, rather than random access processing. It sets an internal index file pointer to a limbo position just ahead of the first record in the index file. A subsequent call to NextKey will fetch the first key in the index file, or a call to PrevKey will fetch the last key in the file.

A data file may have up to 65,535 records, but an index file may have only **Maxint** records, or 32,767. This effectively limits the size of data files to 32,767 records in an indexed system. I have not been able to determine from reading the source code whether Turbo Access could be modified to allow 65,535 keys in an index file.

A record may be up to 65,535 bytes long. This is a constraint of the 8088's register system; a data item cannot be larger than a data segment. Of more interest is the fact that a data record must be at least 8 bytes long. This is to allow Turbo Access to mark a deleted record as deleted (and in fact to forge a linked list of deleted records) without tinkering with the record numbers.

A key string may be between 1 and 255 bytes in size. The number of items in a page must be between 4 and 254.

Testing the speed of a system like this is difficult. Supplied with the Turbo Toolbox is a 1,600-line sample database program that uses most of the routines listed above. A sample database with twenty 342-byte records is included. A search-and-fetch for any given record is only a matter of one short grind of the floppy disk. I had neither the time nor the intestinal fortitude to synthesize a database of tens of thousands of records. However, some pencil-pushing with the Turbo Access specs convinced me that any floppy-based data base with useful-sized records (100 bytes minimum) can be completely searched in four disk accesses or less.

(This might not be the case in 8-bit implementations of Turbo Pascal, with much less RAM available for page buffering. This whole discussion assumes that a 256KB IBM PC is being used.)

An enormous database with upwards of 50,000 records would have to exist on a hard disk, and hard disks are so much faster than floppies that even seven or eight disk accesses would still be faster than three or four on my floppy-based PC.

The conclusion, on speed at least, is that given typical PC, XT, and AT hardware, the response time for any database Turbo Access can handle will be essentially instantaneous (meaning less than one second.)

TURBO SORT

A generalized sort in Pascal is not easy to do. Pascal expects to know array bounds and all data types at compile time, so single-purpose sort routines are easy enough to write; but writing a routine that can sort any given array of any type of data is close to impossible.

Turbo Sort, the second utility in the Toolbox, is not a fully generalized sort package, but it comes as close as any Pascal compiler is likely to allow.

Like Turbo Access, Turbo Sort is a collection of procedures and functions in source form. To use it, however, the user must write three interface routines:

- A routine to get records from somewhere and hand them to Turbo Sort
- A routine to compare two records and determine the larger one in sort order
- A routine to accept the list of sorted records from Turbo Sort and put them back wherever you want them.

As these routines are usually four- or five-liners, they are not an odious burden. However, to sort several different data types within the same program, each data type to be sorted has to have its own mechanism within each of the three custom procedures, selected by a conditional switch of some kind. That's awkward, but that's Pascal.

Turbo Sort uses the Quicksort algorithm, and does all its sorting in heapspace. The sort will work most quickly if all items to be sorted can fit in RAM at once; but when that is not possible Turbo Sort will sort the file in chunks and use the disk as temporary storage. This is an especially attractive feature for 8-bit computers and PC compatibles with small (128K or less) RAM systems. On the AT it buffered large sort files to the hard disk and worked quickly considering the size of the file involved. (See table 1.)

Speed is about the only way to test a sort procedure. My timings include the time it takes to read a file in from disk and write it out again, since those two facets of sorting disk files are inescapable and may involve a considerable (if not major) fraction of the time spent to accomplish the sort.

I sorted files of two kinds of records: Random, generated by Turbo Pascal's random number generator, and ordered, which start with a record like AAAAAAAAAA, followed by AAAAAAAAAB, followed by AAAAAAAAAC and so on. These are the two extremes of the distribution of data in a file, and provide a

best case/worst case test for the speed of a sort method. The results of the time test are in table 1.

The user guide gives some beefy examples of Turbo Sort, including an application involving multiple keys.

Turbo Sort surprised me by using a feature of Turbo Pascal that I had never seen before: untyped VAR parameters. It is possible to pass a VAR parameter to a function or procedure without a data type. Any data type may thus be passed to the procedure in an untyped parameter. This parameter is incompatible with all other data types and thus cannot be assigned to or from like an ordinary variable, but its address may be used to access the data contained in the actual parameter without knowing its type. This is in part how Turbo Sort can sort an array of data items without knowing at compile time what the type of the data item is.

Untyped VAR parameters ought to be used sparingly, but their use in a generalized sort system is a good example of how Turbo Pascal was designed to circumvent some of the built-in limitations of the Pascal language definition.

TABLE 1: *Turbo Sort Benchmarks*

Type of File	PC/ Floppy	PC/ RAM disk	PC/ Fixed	AT/ Floppy	AT/ RAM disk	AT/ Fixed
1,000 by 25 Random	00:54	00:39	00:46	00:30	00:15	00:16
1,000 by 25 Ordered	00:54	00:38	00:36	00:28	00:12	00:13
16,385 by 25 Random	*	*	16:24	*	*	05:24
16,385 by 25 Ordered	*	*	*	*	*	04:10

**Media inadequate for file size*

Turbo sort uses the Quicksort algorithm and does its sorting in heapspace. It works best when all items to be sorted are in RAM.

INSTALLATION SYSTEM

Among the vendors of computer languages, Ginst is a truly original idea, and an excellent one: A means of creating an install program for application programs written in Turbo Pascal. On the Turbo Pascal distribution disk is a program through which Turbo Pascal itself can be installed on many different screen- and keyboard-incompatible MS-DOS computers. If a similar install program could be created for applications programs, a developer could design a program on his favorite computer and sell it for *any* MS-DOS machine, and still use full-screen CRT controls and special keys on the keyboard. The Ginst program is just such a utility. Sadly, it didn't work out as I had hoped.

My test case was this: I have a friend with a Zenith Z100 computer, which is completely incompatible with the IBM PC in terms of display control. I tried to install the Turbo Sort test program on the Z100 after compiling it on the IBM PC, by using an install program generated on the IBM PC by Ginst.

Creating the install program is no more difficult than invoking Ginst and

answering a few questions involving what to name the install program, which application it is intended to install, and where to put the install program when it is created. The resulting install program, when run, presents you with this single menu:

```
Choose one of the following displays:
  0) Default display mode
  1) Monochrome display
  2) Color display 80x25
  3) Color display 40x25
  4) b/w display 80x25
  5) b/w display 40x25
Which display? (Enter no. or Q to exit):
```

The install program then patches your application program to initialize the PC's display to the selected mode.

But what about the Z100? I expected to find a menu of computers and terminals available from the install program (including the Z100), but no such menu exists. I think I know why.

The IBM PC ROM BIOS controls the text display by using software interrupt 10H, as explained in the IBM PC Technical Reference. Most display-incompatible MS-DOS machines like the Z100 and Victor 9000 control their displays by passing control characters and escape sequences to a "console driver" portion of their BIOS. The console driver sees such characters as special cases of characters simply written to the screen. The control characters are snatched off the character stream heading for the screen and interpreted specially. One sequence may clear the screen; another puts the cursor at a specific X,Y; yet another clears an individual line, and so on.

As far as I can tell, Ginst is incapable of crossing the no-man's land between these two CRT-control systems. A program compiled with the IBM PC version of Turbo cannot be installed to operate on a machine that uses a console driver to control its CRT. A program compiled on a machine with a console driver cannot be installed to work correctly on the PC with interrupt 10H.

Or, more tersely, you can develop software on the Zenith Z100 and install it on the Victor 9000, but not on the IBM PC. Likewise, software developed on the PC can be installed *only* on compatibles that correctly interpret software interrupt 10H. And in those cases, no installation is really necessary unless a user means to change between the PC's 40- and 80-column display modes.

This is an aching disappointment. The compatibility gap between the software interrupt system and the console driver system is a big one.

Borland suggested that by using IBM's ANSI.SYS console driver and the "generic" MS-DOS Turbo Pascal (which assumes a console-driver screen control system) software developed on the IBM PC could be installed on screen-incompatible machines. This is true; however, the ANSI.SYS driver is a console driver overlayed upon the ROM BIOS INT 10H code, and is far too slow to be useful. Speedy CRT updates require that a system be optimized toward one method or the other. The Ginst program could conceivably plug two different flavors of CRT-control machine code into the various MS-DOS machines, one for console drivers and one for software interrupts. But it does not.

Ginst is thus useful mostly for transferring programs between one console driver system and another with different control codes. This is a valid use, and will make it easier to develop software on PC-incompatible (but otherwise excellent) machines like the Zenith and the Victor. Perhaps a future release of the Toolbox will knit the gap together again.

RELIABILITY

Turbo Pascal is the most nearly bug-free compiler I have ever used, and the Turbo Toolbox comes close in quality. I

found only one bug in my two weeks of tinkering: Turbo Sort, which can theoretically sort files of up to 32,767 (**Maxint**) records, will crash while sorting any file of more than 16,385 records. Since 16,385 is half of **Maxint** plus one, my intuition told me that there was an integer overflow problem somewhere. Once alerted to the bug, Borland took only a few days to locate the problem and provided a patch for Sort.box. This patch will need to be made to all copies of the Turbo Toolbox up to approximately serial number 13,000. Since it is a patch to source file, the user needs to add only Sort.box into the Turbo text editor and locate the only incidence of the following: M : = (I + J) **div** 2. In this line change **div** 2 to **shr** 1. This line of code occurs in the vicinity of line 231 of Sort.box.

While I was adding records to the Turbo Access sample database (on an extra and expendable copy of the disk) I got devilish and hit the power switch. As expected, I lost the records I had input, but surprisingly had no trouble with corrupted data sets. All data sets appeared to be in the same condition they had been when I began entering records. I suspect that if I had entered enough records to force Turbo Access to begin rearranging the shape of the B-tree I would have had bigger trouble. It is yet another reason to save often.

Documentation is where the Turbo Toolbox falls short. Its user guide is outwardly flashy (slick four-color cover) and neatly typeset in the Borland style, but it is otherwise a terrible mess. Someone obviously wrote it in a hurry and never bothered to proof it.

- The cover does not mention Turbo Access, but speaks of Turbo ISAM instead. No mention of ISAM inside.
- Here and there we see mention of B + trees.
- The source include file for Turbo Access is sometimes called Dtaman.box and sometimes Access.box. On the disk it is Access.box. I thought for awhile that something was missing. It was not until I had read the entire manual that I saw that the two were in fact the same file.

These are only the most blatant and confusing examples; there are a host of misspelled words and minor textual goof-ups.

Another area in which Borland missed the boat is in not using diagrams. As structured, two-dimensional entities, B-trees lend themselves well to simple line diagrams. A newcomer who has never studied data structures will have a real picnic visualizing a B-tree from the highly technical prose descriptions. (A tutorial on B-Trees appears on page 78 of this issue.)

While not a breakthrough product the way that Turbo Pascal itself was, the Turbo Toolbox is solid enough and useful enough to come recommended, especially if you write file-management programs in Turbo Pascal.

By far, the best part of the package is Turbo Access. It is easily worth the entire $49.95. In a way, the excellence of Turbo Access makes Turbo Sort less necessary—there is little need to sort a file when you can maintain a set of key files equivalent to having the same file sorted a different way for each key file.

Ginst may not be especially useful, since it does not allow terminal installation on the IBM PC except through ANSI.SYS, an inadequate console driver.

The documentation is hasty and sloppy. On the other hand, I have seen much worse software packages costing a whole lot more. One would like to have good software, a low price, and competent documentation. But two out of three isn't bad.

Jeff Duntemann's writing on languages and language products frequently appears in these pages. He is a programmer for Xerox Corporation in Rochester, New York.

The Power of Turbo Pascal

Tree-structured directories and removable windows illustrate the extension of Turbo Pascal's capabilities to other programs.

MICHAEL COVINGTON

With Turbo Pascal, the user can access exotic features of the machine or the operating system.

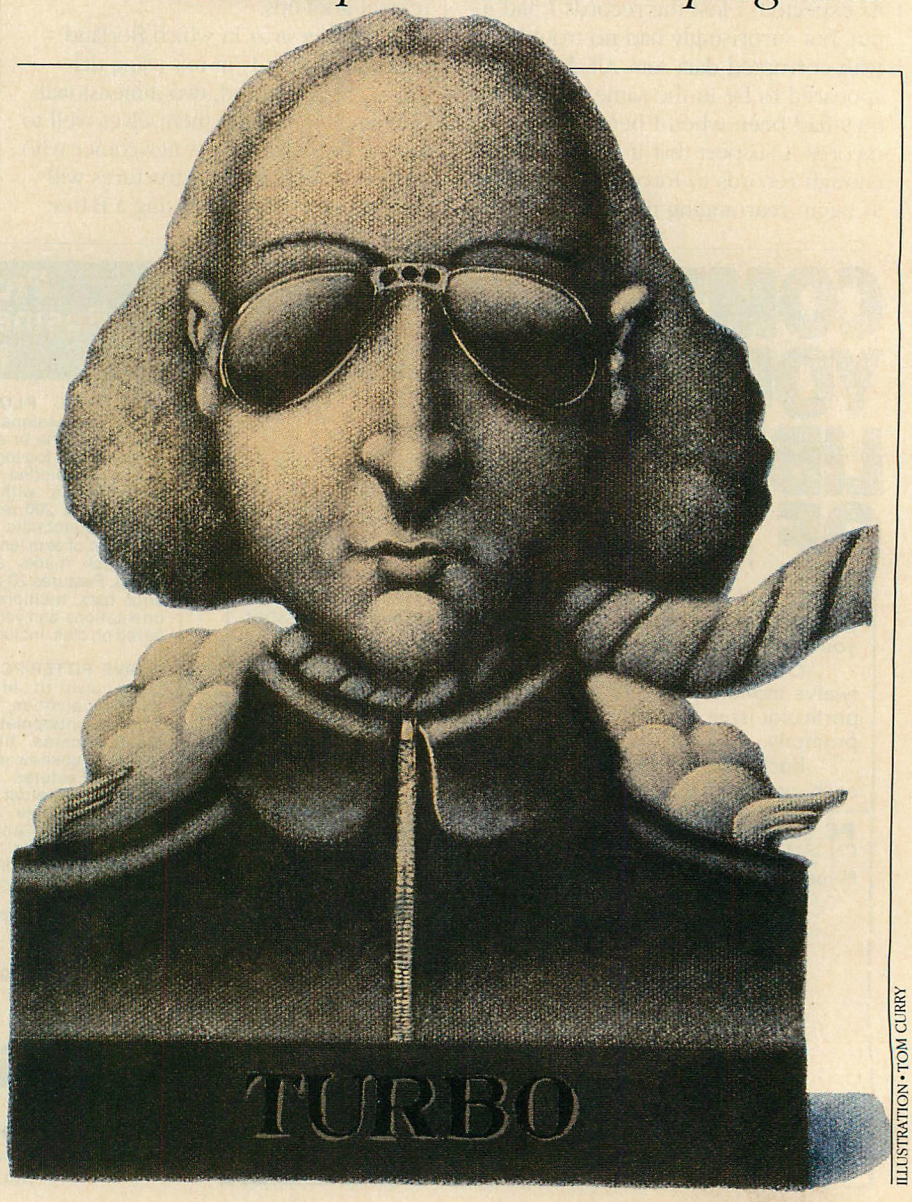


ILLUSTRATION • TOM CURRY

Practically anything assembly language can do Turbo Pascal can do more easily. This compiler, produced by Borland International, is one of the best programming tools presently available for the PC. It is certainly the cheapest and fastest, and it produces the most compact object code. The built-in, full-screen editor allows the user to correct syntax errors at the instant the compiler finds them.

With Turbo Pascal, the user can access exotic features of the machine or the operating system. Variables can be placed at specific memory addresses, data can be picked apart into bytes or even bits, and any interrupt can be performed with a subroutine call.

Using Turbo Pascal 2.0B under PC-DOS 2.1, I developed the following subroutines, which can add power to programs by accessing features of the PC that most languages cannot. These subroutines can be used in Turbo Pascal programs or adapted for use with other Pascal compilers or with other languages such as C or assembly.

These routines have been designed in such a way that using them does not require knowledge of how they work; just copy the listing into the program at the appropriate place and call it in the specified way. In the Turbo Pascal editor, Ctrl-K Ctrl-R inserts another file into the file being edited; this is a convenient way to bring in a subroutine package because the subroutines can be modified or deleted as needed. Alternatively, the $I compiler directive can be used to read subroutines from another file at compile time. No name conflicts exist between the packages, and globally recognized names are identified at the beginning of each listing.

On the basis of DOS documentation, I have indicated which routines can be expected to run under earlier versions of DOS than 2.1, but I have not actually conducted tests to confirm this.

The compiled object programs that Turbo creates are .COM files; they can be invoked by typing their file names (minus the .COM suffix) as commands. For example, a file called **a:myprog.com** can be invoked by typing

```
A> myprog
```

The ability to pass parameters to programs from the command line is useful. That is, by typing

```
A> myprog hello
```

the user ought to be able to pass the string 'hello' to his program. The procedure **getparm** (listing 1) does this. It takes one argument, a variable of type **parmtype** (for example, **string**[127]), and returns, in it, the next parameter waiting to be read from the command line or an empty string if there is no parameter waiting. For example, if a program consists of listing 1 plus

```
var x: parmtype;
begin
 repeat
   getparm(x);
   writeln(x)
 until x = ' '
end.
```

and it is invoked with the command

```
A> myprog alpha beta gamma
```

it will print the following, then stop:

```
alpha
beta
gamma
```

If the main program were to keep calling **getparm** after all the parameters had been retrieved, it would get an empty string each time.

Each parameter is returned as a varying-length string with no leading or trailing blanks. Thus, parameters representing file names, such as **'a:mydata.txt'**, can be used as arguments to the built-in **assign** procedure. Also, the built-in procedure **val** allows conversion of strings such as '25' to numbers.

Getparm cannot retrieve I/O-redirection parameters such as >LPT1:— DOS edits these out of the parameter list before passing it to the program. This is as it should be, since redirection is handled entirely by DOS and is supposed to be transparent to the user's software. As it happens, Turbo Pascal handles files by means of the original, DOS 1.0-compatible set of operating system services, which are not affected in any way by I/O redirection.

To obtain the parameter list, **getparm** relies on a variable assigned to a particular memory location. When DOS loads a program, it puts the first byte of the program at offset 100H in the code segment. The lower addresses, CSEG:0H to CSEG:FFH, contain a work area known as the *program segment prefix*, and in it, at CSEG:80H, is a 128-byte area containing the command line parameters. The first byte is interpreted as a number indicating how many of the following 127 bytes are significant.

This happens to be exactly how Turbo Pascal represents a varying-length string, so all the user has to do is define the parameter list as a string whose byte 0 (containing the length indicator) is at CSEG:80H. This string is then eligible for all the normal string operations of Turbo Pascal; it is straightforward to delete leading blanks and parameters from the string as they are retrieved, so that the string gets shorter after each call until it is empty.

Note, incidentally, that it is legal to refer to **parms**[1] even when the value of **parms** is a string of zero length. **Getparm** does this in the condition tests of both **while** loops. In Turbo Pascal, a varying-length string is represented as an array of characters numbered beginning with 0. Byte 0 (for example, **parms**[0]) contains a numeric value giving the length of the string; the string itself begins at byte 1. Thus, a variable declared as **string**[10] is in reality an **array**[0..10] **of char**, and all of the elements of the array exist at all times; element 0 indicates how many of them contain valid data.

Getparm should work with any version of MS-DOS or PC-DOS. If it is adapted for use with another compiler, the user's program may need to retrieve all its parameters before it opens any files; some types of programs overwrite the parameter area when a disk file is opened, although Turbo Pascal programs apparently do not.

SETTING THE CLOCK

Applications programs often need to determine the date and time in order to include them in print-out headings or screen displays; it is also useful to be able to set the date and time. These are easy operations in BASIC, but, surprisingly, Turbo Pascal does not provide procedures to perform them.

Listing 2 fills this gap. The functions **date** and **time** return eight-character strings of the forms '09/20/84' and '12:31:19', so that if the statement

```
writeln(DATE, ' ',TIME)
```

is executed, the computer will display the date and time. The strings can be manipulated to display the date and time in other formats or even perform computations on them.

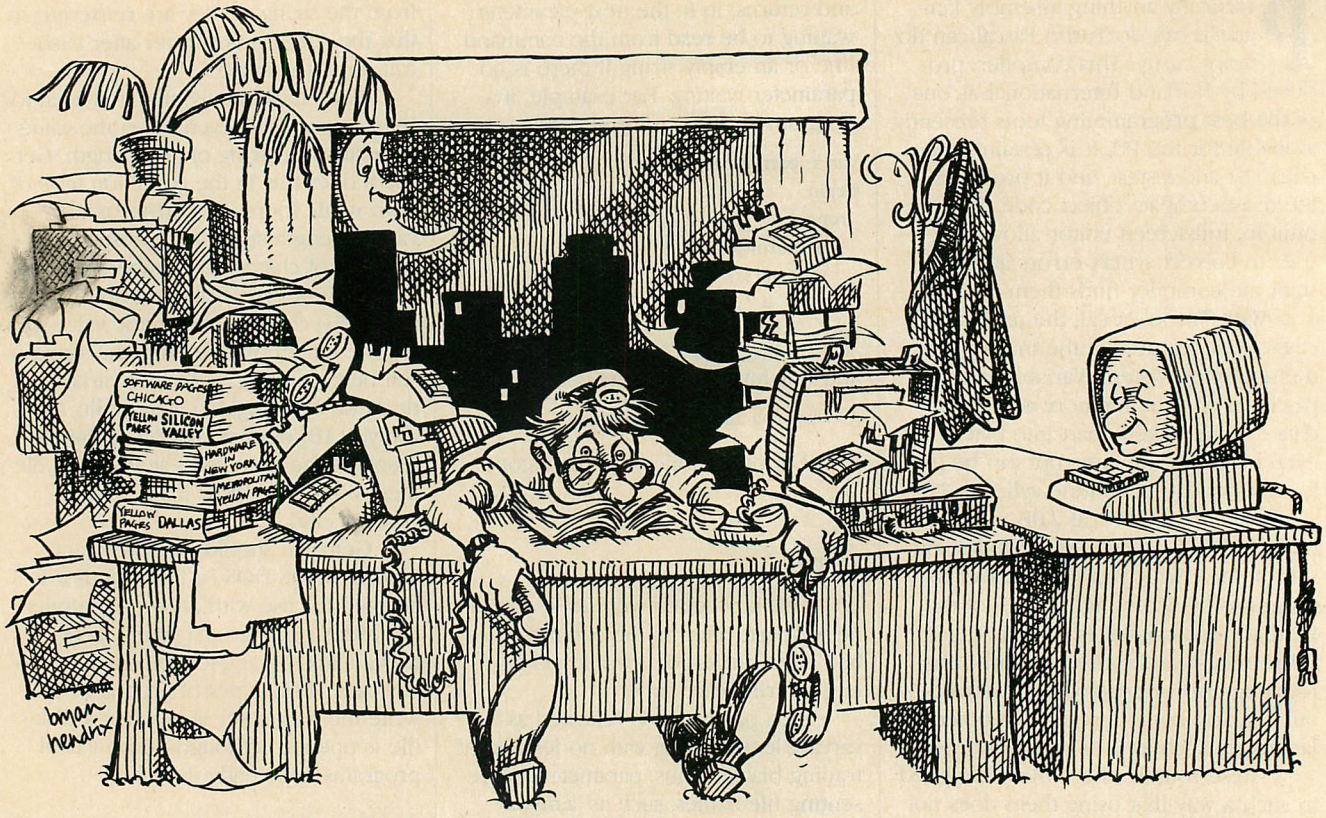
The procedures **setdate** and **settime** do the opposite; they accept strings in the same format as used by **date** and **time**, and set the clock accordingly. Their arguments can be either constants or variables; both

```
setdate('09/20/84')
```

and

```
var x: string[8];
...
writeln('What is the date?');
readln(x);
setdate(x)
```

are valid. **Setdate** and **settime** halt the program if they receive an invalid argument, but this can be changed if de-



Spending more time searching for the technology...than using it?

DATA SOURCES systematically organizes all your DP options!

No other resource puts over 20,000 systems and applications software packages...over 17,000 hardware and data communications products...and some 9,000 company profiles instantly at your fingertips.

DATA SOURCES is conveniently indexed to help you locate the products you seek quickly...find others you may not be aware of...and determine their capability and compatibility at a glance. You'll use...

...The Product Index, with its complete listing of every product covered by DATA SOURCES, to lead you to the larger listing quickly.

...The Company Profile Index, to help you identify all the product areas in which a firm is currently involved easily.

...The Company/Product Index, to find each vendor's full range of products within each equipment category instantly.

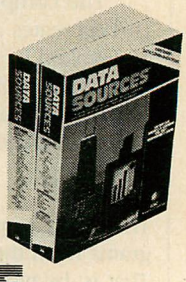
...The Software Index, to help you access specific packages, even if you're not sure of the author or product classification.

And DATA SOURCES' handy comparison charts let you review similar products by multiple manufacturers for key criteria, including compatibility and price.

PLUS...in-between quarterly editions, subscribers have access to the exclusive DATA SOURCES HOTLINE for information on product location assistance.

JUST CALL (800) 227-1617 ext 251 TODAY! (In California, call (800) 722-3543 ext. 251)

We'll send your first 2-volume edition on a 30-DAY APPROVAL basis and enter your year's subscription to DATA SOURCES at just $150...$60 off the regular cover price...FOR 4 quarterly editions.



DATA SOURCES T325

Leading the Industry in Information Management

P.O. Box 5845, Cherry Hill, N.J. 08034

CIRCLE NO. 202 ON READER SERVICE CARD

sired. However, at the operating system level, invalid arguments do no harm; they leave the clock unaffected.

Each of these routines uses the type definitions at the beginning of the listing, but the routines do not call each other; after listing 2 is embedded in a program, any unused routines that are present can be deleted.

Date, **time**, **setdate**, and **settime** work by performing interrupts to access DOS-provided interrupt service routines. All DOS services used here are accessed through interrupt 21H; the contents of the AH register determine which service is performed; the other registers pass information to and from the interrupt service routine.

Turbo Pascal has a built-in procedure **intr** that performs any interrupt. It takes two parameters: a number identifying the interrupt and a record of values to be loaded into the registers. The type **regtype** defined at the beginning of the listing establishes the structure of this record and associates the usual register names with its components; **intr** copies the record into the registers, performs the interrupt, and copies the registers back into the record so the calling program can retrieve them.

Note that each register (for example AX) is treated as a two-byte integer. When it is necessary to examine half-registers (for example, AH and AL), the built-in functions LO and HI give the values of the low and high halves respectively; **date** and **time** use this technique extensively. One way to load values into the low and high bytes of a register is to multiply the high value by 256 (100H), add it to the low value, and assign the result to the whole integer; the procedures **setdate** and **settime** do this in several places.

The interrupt services are described in detail in the *DOS Technical Reference*. In brief, the function **date** loads 2AH into AH to identify the service being requested; performs the interrupt; retrieves various registers; converts the numbers in them into strings using the built-in **str** procedure; and assembles the properly formatted string. **Time** is quite similar. **Setdate** and **settime** do the opposite: they convert strings into numbers using **val**, load the appropriate values into registers, perform the interrupt, and, last of all, check whether either **val** or the interrupt itself detected an error. Postponing the error-checking until the last step is legitimate because invalid arguments are caught by the interrupt service routine and have no effect on the clock. **Date**, **time**, **setdate**, and **settime** should work under any version of DOS.

TREE-STRUCTURED DIRECTORIES

The big attraction of DOS 2.0 and later versions is, of course, that it gives the user the ability to create subdirectories and organize files into tree-like structures. A file is then referred to, not just by a name, but by a path, such as the following example

```
'a:\alpha\gamma\myprog.pas',
```

consisting of the drive letter and a chain of directory names leading down to the actual location of the file. If it begins with "\" (after the drive letter, if any), the path starts at the topmost root directory of the disk; otherwise it starts at the current default directory, where files referred to by plain names (without paths) are assumed to be.

For compatibility with later versions of DOS, Turbo Pascal accesses files by file name only, and so can reach only the files in the current directory. The Turbo environment provides a command (the letter A) to move from one directory to another, but does not provide a way to make such moves from within the program.

Determining and setting the date and time are easy operations in BASIC, but, surprisingly, Turbo Pascal does not provide built-in procedures to perform them.

Listing 3, treedir.pas, remedies the problem by providing a full range of directory manipulation routines. Unfortunately, since Turbo Pascal is still tied to the DOS 1.0-style file system, files that are outside the current directory cannot be accessed—that is to say, files that are in different directories cannot be open at the same time.

But any file can be accessed by going to the appropriate directory, working with it, and then coming back to the original directory. In addition, new directories can be created and deleted, and the user can move files from one directory to another.

The package contains 11 routines; each contains comments indicating which of the other routines it requires, if any. Naturally, all the routines require the type definitions at the beginning. (Note that **rtype** here is equivalent to **regtype** in the date and time package; a different name is used to prevent conflicts if both packages are embedded in the same program.)

The procedures **xxdiskerr** and **xxpatherr** write error messages for many of the other routines. The remaining procedures manipulate the file system by calling DOS services through interrupt 21H. In most cases, the services are provided only by DOS version 2.0 and later, but **currentdrive** and **chdrive** are exceptions; they should work under all versions of DOS.

Currentdrive is a function that returns the current drive letter followed by a colon. For instance, if drive A: is currently selected, and the statement

```
writeln(currentdrive)
```

is executed, the computer will print 'A:'. **Chdrive** is the inverse operation; it accepts an argument of the same form, upper or lower case, with or without the colon. For example,

```
chdrive('A:')
```

changes the default drive to A:.

Diskspace returns the number of bytes available on a particular disk. It takes an argument like that of **chdrive**. For example:

```
writeln(diskspace('A:'):10:0,
    'bytes free on drive A.')
```

The ':10:0' specification in the **writeln** statement is necessary because the value of **diskspace** is of type **real**. Naturally, it can be used in calculations—for example, it can be used to determine if a particular disk has enough space available to create a file.

Currentdir is like **currentdrive** except that it returns the path to the current directory on the specified disk. It, too, takes a drive letter as a parameter. The returned value is the complete path, beginning with a backslash but without the drive letter. For example, the value of **currentdir('B:')** might be **'\myprogs\wrkspace'**. Naturally, it ends with the name of a directory, not the name of a file. The main use for **currentdir** is to determine the default directories in effect at the beginning of the program so that, after altering them to suit the program's purposes, the user can restore them to the original values.

The three routines **chdir**, **rmdir**, and **mkdir** are like the DOS commands with the same names: **chdir** changes the current directory, **rmdir** deletes a directory (which must not contain any files), and **mkdir** creates a directory. Each of them takes a single parameter, a string of up to 63 characters, that is interpreted the same way as the parameter of the corresponding DOS command.

Additionally, the DOS calls needed for the three operations are identical for the value of the AH register, so all three of the routines use the procedure **xxdir** to format the parameter and perform the interrupt.

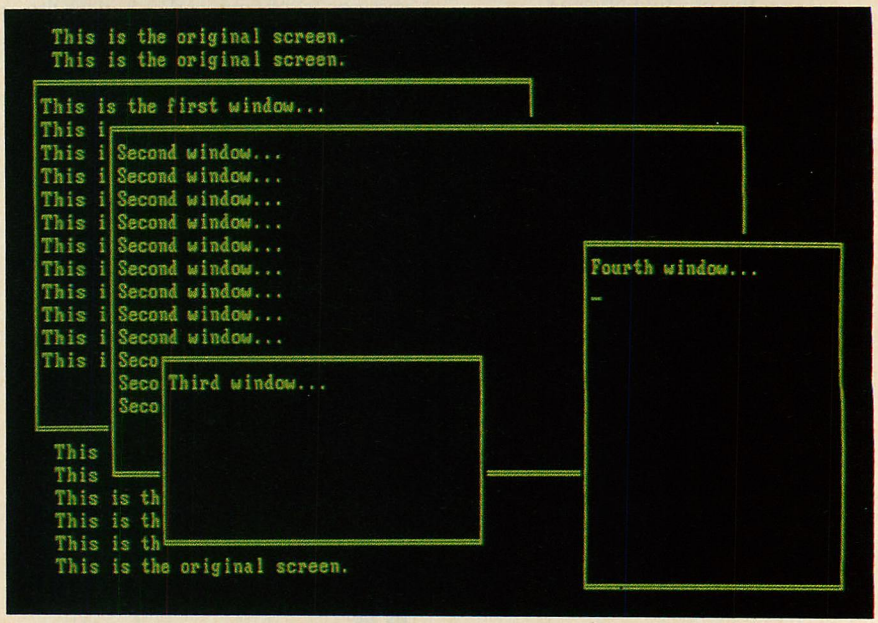
The **rename** procedure is similar to the corresponding DOS command, but more powerful. In DOS, **rename** can only change the name of a file; this **rename** can also move a file from one directory to another. For example,

```
rename('a:\alpha\myprog.pas',
       '\beta\myprog.pas')
```

moves **myprog.pas** from directory **alpha** to directory **beta**. It still cannot move a file from one disk to another, however; if a drive letter is in the second parameter, it is ignored.

The DOS interrupt services that use path names represent them as "ASCIIZ strings"—IBM's term for a string of bytes ending with ASCII code 0. In order to convert a Turbo Pascal varying-length string into an ASCIIZ string, simply add a **chr(0)** at the end and then reckon the beginning of the ASCIIZ string from byte 1 of the Turbo Pascal string (rather than byte 0, which is Turbo's length indicator). This is done in procedures **xxdir** and **rename**. The built-in functions **seg** and **ofs** return the segment and offset, respectively, of the specified byte; these are loaded into registers to tell the DOS service where the string is stored.

PHOTO 1: *Nested Windows*



The maximum depth of nesting is set by the constant **maxwin** and is limited only by available memory.

Converting an ASCIIZ string back to a Turbo Pascal string, as in procedure **currentdir**, is done by ignoring the length indicator (which has not yet been set to a meaningful value), searching through the string for the **chr(0)** that marks the end, and then putting the right value into byte 0 to indicate

length. As noted above, it is legitimate to refer to all of the bytes of the string whether or not the length indicator says those bits are used.

REMOVABLE WINDOWS

Turbo Pascal allows the user to define windows in such a way that screen output and scrolling to a particular area of the screen can be confined. When this is done, the window cannot be easily taken away to restore whatever was behind it—which means that windowing in Turbo is different from windowing in other environments.

The routines in listing 4 implement true removable windows in Turbo Pascal. Each time a window is opened, the entire contents of the screen are saved, along with the cursor position and window dimensions. When that window is closed, the previous contents of the screen are restored. Windows can be nested (photo 1); the maximum depth of nesting is set by the constant **maxwin** and is limited only by available memory, since it takes slightly more than 4KB to store each image of the screen. The demonstration program in listing 5 tests the whole package.

Since the screen contents are saved and restored by copying the contents of video memory, this package will work only on computers that use the same memory addresses as the IBM PC. I tried the alternative—using BIOS interrupts to obtain the contents of each screen location one by one—and found it intolerably slow. Demanding though it may be of the hardware, this package makes no assumptions about the operating system; it should run under any version of DOS or even CP/M-86.

The procedure **initwin** must be called before any of the others in order to initialize global variables. The procedure **mkwin** creates a window and it takes four parameters. For example, the following statement

```
mkwin(5,2,30,15)
```

creates a window that extends from column 5, line 2 (numbered from the upper left) to column 30, line 15. The actual working area is two units smaller in each direction, since two columns and two lines are occupied by the border. The cursor starts out at the upper left corner of the window.

Rmwin is used to remove the window. It takes no parameters. Internally, **mkwin** uses the procedure **boxwin** to draw the border and define a restricted scrolling area (what Turbo Pascal calls a window). **Boxwin** can be used by itself—but **boxwin** and the built-in procedure **window** should not be called from within a program that uses **mkwin** and **rmwin**, since **rmwin** assumes that there are no windows that were not created by **mkwin**.

The window package works only in text mode, not graphics mode, and only on display page 0 (the constant display page unless BIOS calls have changed it). Also remember that while the window feature saves cursor position and window size, it does not save or restore the currently selected video intensity, text color, or display width.

These routines are only a sample of what can be done with Turbo Pascal; in general, a Turbo Pascal program can make the PC do anything it is capable of doing. A programming language that provides such a combination of power and convenience is difficult to find; I, therefore, expect to be using this one for a long time to come.

Michael Covington does research on artificial intelligence and supercomputer applications at the University of Georgia.

LISTING 1: getparm.pas

```
{ Turbo Pascal procedure to retrieve command line parameters }
{ Copyright 1984 Michael A. Covington }


type parmtype = string[127];


procedure getparm(var s:parmtype);


   { Returns first available parameter from DOS command  }
   { line and removes it so next parameter will be       }
   { returned on next call.  If no more parameters are   }
   { available, returns a null string.                   }


var parms: parmtype absolute CSEG:$80;
begin
  s:='';
  { parms[1] exists even when length is zero }
  while (length(parms) > 0) and (parms[1] = ' ') do
    delete(parms,1,1);
  while (length(parms) > 0) and (parms[1] <> ' ') do
    begin
      s:=s+parms[1]; delete(parms,1,1)
    end
end;
```

LISTING 2: datetime.pas

```
{ Turbo Pascal routines to read and set date and time }
{ Copyright 1984 Michael A. Covington }


{ Each routine requires the following type definitions }
{ but does not require the other routines.             }

type datetimetype = string[8];
     regtype      = record
                      ax,bx,cx,dx,bp,si,di,ds,es,flags: integer
                    end;


function date: datetimetype;
  { Returns current date in form '08/31/84'. }
var  reg:     regtype;
     y,m,d,w: datetimetype;
     i:       integer;

begin
  reg.ax:=$2A00;
  intr($21,reg);
  str(reg.cx:4,y);
  delete(y,1,2);
  str(hi(reg.dx):2,m);
  str(lo(reg.dx):2,d);
  w := m + '/' + d + '/' + y;
  for i:=1 to length(w) do if w[i]=' ' then w[i]:='0';
  date:=w
end;


function time: datetimetype;
  { Returns current time in form '08:13:59'. }
var  reg:     regtype;
     h,m,s,w: datetimetype;
     i:       integer;

begin
  reg.ax:=$2C00;
  intr($21,reg);
  str(hi(reg.cx):2,h);
  str(lo(reg.cx):2,m);
  str(hi(reg.dx):2,s);
  w := h + ':' + m + ':' + s;
  for i:=1 to length(w) do if w[i]=' ' then w[i]:='0';
  time:=w
end;
```

```
procedure setdate(x:datetimetype);
  { Sets date.  Accepts string in format '08/31/84'. }
var  reg:           regtype;
     rh,rl,c1,c2,c3: integer;
begin
  reg.ax:=$2B00;
  val(x[1]+x[2],rh,c1);  { month goes in DH }
  val(x[4]+x[5],rl,c2);  { day goes in DL   }
  reg.dx:=rh*256 + rl;
  val(x[7]+x[8],rl,c3);  { year goes in CX  }
  reg.cx:=rl + 1900;
  if rl<80 then reg.cx:=reg.cx+100;  { 21st century }
  c1:=c1+c2+c3;          { return codes from VAL }
  if c1=0 then intr($21,reg);
  if c1+lo(reg.ax) <> 0 then
    begin
      writeln;
      writeln('Error--Invalid date, ''',x,'''');
      halt
    end
end;


procedure settime(x:datetimetype);
  { Sets time.  Accepts string in format '08:13:59'. }
var  reg:           regtype;
     rh,rl,c1,c2,c3: integer;
begin
  reg.ax:=$2D00;
  val(x[1]+x[2],rh,c1);    { Hours go in CH   }
  val(x[4]+x[5],rl,c2);    { Minutes go in CL }
  reg.cx:=rh*256 + rl;
  val(x[7]+x[8],rh,c3);    { Seconds go in DH }
  reg.dx:=rh*256;
  c1:=c1+c2+c3;            { Return codes from VAL }
  if c1=0 then intr($21,reg);
  if c1+lo(reg.ax) <> 0 then
    begin
      writeln;
      writeln('Error--Invalid time, ''',x,'''');
      halt
    end
end;
```

LISTING 3: treedirs.pas

```
{ Turbo Pascal routines for tree-structured directories }
{ Copyright 1984 Michael A. Covington }


{ Requires MS-DOS or PC-DOS 2.0 or higher, except as noted. }


{ All the routines require these type definitions.          }
{ However, except as noted, they do not require each other. }


type pathtype  = string[63];
     drivetype = string[2];
     rtype     = record
                   ax,bx,cx,dx,bp,si,di,ds,es,flags: integer
                 end;


procedure xxdiskerr(x:drivetype);
begin
  writeln('Error -- Invalid disk drive, ''',x,'''');
  halt
end;


procedure xxpatherr(x:pathtype);
begin
  writeln('Error -- Invalid path, ''',x,'''');
  halt
end;


function currentdrive: drivetype;
```

```
  { Returns designator for current default drive, e.g., 'A:'. }
  { Works under DOS version 1. }
var  w:   drivetype;
     reg: rtype;
begin
  reg.ax:=$1900;
  intr($21,reg);
  w:='A:';
  w[1]:=chr(ord(w[1])+lo(reg.ax));
  currentdrive:=w
end;


procedure chdrive(x:drivetype);
  { Chooses a new default drive. }
  { Parameter can have the form 'A:', 'A', 'a:', or 'a'.    }
  { Works under DOS version 1.  Requires XXDISKERR, above.  }
var  reg: rtype;
begin
  reg.ax := $0E00;
  reg.dx := ord(upcase(x[1])) - ord('A');
  intr($21,reg);
  if (reg.dx < 0) or (lo(reg.ax) < lo(reg.dx)) then xxdiskerr(x);
end;


function diskspace(x:drivetype): real;
  { Returns number of bytes available on specified disk.   }
  { Parameter as for CHDRIVE.  Requires XXDISKERR, above.  }
var reg: rtype;
begin
  reg.ax := $3600;
  reg.dx := 1 + ord(upcase(x[1])) - ord('A');
  intr($21,reg);
  if reg.ax = $FFFF then
    xxdiskerr(x)
  else
    diskspace := (256.0*hi(reg.dx)+lo(reg.dx)) * reg.ax * reg.cx
end;


function currentdir(x:drivetype): pathtype;
  { Returns full path to active directory on specified drive, }
  { including backslash at beginning, not including drive     }
  { designator.  Parameter as for CHDRIVE.                    }
  { Requires XXDISKERR, above.  }
var  w:   pathtype;
     reg: rtype;
     i:   integer;
begin
  { Get current path }
  reg.ax:=$4700;
  reg.dx:=1 + ord(upcase(x[1])) - ord('A');
  reg.ds:=seg(w[1]);
  reg.si:=ofs(w[1]);
  intr($21,reg);
  if (reg.flags and 1) > 0 then xxdiskerr(x);

  { Turn it into a Turbo string }
  i:=1;
  while w[i]<>chr(0) do i:=i+1;
  w[0]:=chr(i-1);
  for i:=1 to length(w) do w[i]:=upcase(w[i]);

  currentdir := '\' + w
end;


procedure xxdir(x:pathtype; k:integer);
  { Executes CHDIR, MKDIR, and RMDIR requests. }
  { Requires XXPATHERR and CURRENTDRIVE, above. }
var w:   pathtype;
    reg: rtype;
begin
  w := x + chr(0);
  if w[2] <> ':' then  { add drive designator }
    w := currentdrive + w;
  reg.ax := k;
  reg.ds := seg(w[1]);
  reg.dx := ofs(w[1]);
  intr($21,reg);
  if (reg.flags and 1) > 0 then xxpatherr(x)
end;


procedure chdir(x:pathtype);
```

```
  { Equivalent to CHDIR command in DOS.                }
  { Requires XXDIR, XXPATHERR, and CURRENTDRIVE, above. }
  { Caution!  Do not leave a directory }
  { if you have files in it open.      }
begin
  xxdir(x,$3B00)
end;


procedure rmdir(x:pathtype);
  { Equivalent to RMDIR command in DOS.                }
  { Requires XXDIR, XXPATHERR, and CURRENTDRIVE, above. }
begin
  xxdir(x,$3A00)
end;


procedure mkdir(x:pathtype);
  { Equivalent to MKDIR command in DOS.                }
  { Requires XXDIR, XXPATHERR, and CURRENTDRIVE, above. }
begin
  xxdir(x,$3900)
end;


procedure rename(x,y:pathtype);
  { Renames a file; unlike the DOS RENAME command,  }
  { both parameters of this command are full paths. }
  { The paths need not be the same, allowing a file }
  { to be moved from one directory to another.      }
  { First parameter can specify a drive; any drive  }
  { letter on the second parameter is ignored.      }
var  wx,wy:  pathtype;
     reg:    rtype;
begin
     wx := x + chr(0);
     wy := y + chr(0);
     if wx[2]<>':' then wx := currentdrive + wx;
     reg.ax := $5600;
     reg.ds := seg(wx[1]);
     reg.dx := ofs(wx[1]);
     reg.es := seg(wy[1]);
     reg.di := ofs(wy[1]);
     intr($21,reg);
     if (reg.flags and 1) <> 0 then
       begin
         writeln('Error -- Invalid rename request');
         writeln('      -- From: ''',x,'''');
         writeln('      -- To:   ''',y,'''');
         halt
       end
end;
```

LISTING 4: window.pas

```
{ Turbo Pascal removable window system }
{ Copyright 1984 Michael A. Covington  }


{ Requirements: IBM PC or close compatible.   }
{ Screen must be in text mode, on page 1,     }
{ either mono or color card.                  }


{ Call INITWIN before calling MKWIN or RMWIN. }

const maxwin = 5;  { maximum number of windows open at once }

type imagetype  = array [1..4096] of char;
     windimtype = record
                    x1,y1,x2,y2: integer
                  end;
var
  win: { Global variable package }
    record
      dim:    windimtype;  { Current window dimensions }
      depth:  integer;
      stack:  array[1..maxwin] of
                record
                  image: imagetype;  { Saved screen image }
```

```
                dim:   windimtype; { Saved window dimensions }
                x,y:   integer     { Saved cursor position }
              end
    end;


  crtmode:      byte      absolute $0040:$0049;
  crtwidth:     byte      absolute $0040:$004A;
  monobuffer:   imagetype absolute $B000:$0000;
  colorbuffer:  imagetype absolute $B800:$0000;

procedure initwin;
  { Records initial window dimensions }
begin
  with win.dim do
    begin x1:=1; y1:=1; x2:=crtwidth; y2:=25 end;
  win.depth:=0
end;


procedure boxwin(x1,y1,x2,y2:integer);
  { Draws a box, fills it with blanks, and makes it the current }
  { window.  Dimensions given are for the box; actual window is }
  { one unit smaller in each direction.                         }
  { This routine can be used separately from the rest of the    }
  { removable window package.                                   }
var x,y: integer;
begin
  window(1,1,80,25);
  { Top }
  gotoxy(x1,y1);
  write(chr(213));
  for x:=x1+1 to x2-1 do write(chr(205));
  write(chr(184));

  { Sides }
  for y:=y1+1 to y2-1 do
    begin
      gotoxy(x1,y);
      write(chr(179), ' ':x2-x1-1, chr(179))
    end;



  { Bottom }
  gotoxy(x1,y2);
  write(chr(212));
  for x:=x1+1 to x2-1 do write(chr(205));
  write(chr(190));

  { Make it the current window }
  window(x1+1,y1+1,x2-1,y2-1);
  gotoxy(1,1)
end;



procedure mkwin(x1,y1,x2,y2:integer);
  { Create a removable window }

begin
  { Increment stack pointer }
  with win do depth:=depth+1;
  if win.depth>maxwin then
    begin
      writeln(^G,' Windows nested too deep ');
      halt
    end;



  { Save contents of screen }
  if crtmode = 7 then
    win.stack[win.depth].image := monobuffer
  else
    win.stack[win.depth].image := colorbuffer;

  win.stack[win.depth].dim := win.dim;
  win.stack[win.depth].x   := wherex;
  win.stack[win.depth].y   := wherey;

  { Create the window }
  boxwin(x1,y1,x2,y2);
  win.dim.x1 := x1+1;
  win.dim.y1 := y1+1;    { Allow for margins }
```

```
  win.dim.x2 := x2-1;
  win.dim.y2 := y2-1;
end;


procedure rmwin;
  { Remove the most recently created removable window }
  { Restore screen contents, window dimensions, and   }
  { position of cursor.  }
begin
  if crtmode = 7 then
    monobuffer := win.stack[win.depth].image
  else
    colorbuffer := win.stack[win.depth].image;
  with win do
    begin
      dim := stack[depth].dim;
      window(dim.x1,dim.y1,dim.x2,dim.y2);
      gotoxy(stack[depth].x,stack[depth].y);
      depth := depth - 1
    end
end;
```

LISTING 5: windemo.pas

```
{ Demonstration program for removable window package }


{ --- Insert code from Listing 4 here --- }

var i: integer;

begin
  initwin;
  writeln('Now and every time the action stops,');
  writeln('press ENTER to continue.');
  readln;
  clrscr;
  for i:=1 to 25 do writeln('    This is the original screen.');

  mkwin(3,3,50,18);
  for i:=1 to 15 do writeln('This is the first window...');
  readln;

  mkwin(10,5,70,20);
  for i:=1 to 15 do writeln('Second window...');
  readln;

  mkwin(15,15,45,23);
  writeln('Third window...');
  readln;

  mkwin(55,10,79,25);
  writeln('Fourth window...');
  readln;

  rmwin;   { Remove fourth window }
  readln;

  rmwin;   { Remove third window }
  writeln;
  writeln('We are back in the second window...');
  readln;

  rmwin;   { Remove second window }
  writeln;
  writeln('This is the first window again!');
  readln;

  rmwin;   { Remove first window }
  readln
end.
```

Super assemblers plus the world's largest selection of cross assemblers!

Z-80 Macroassembler $49.50

Power for larger programs! This 2500AD macroassembler includes:

- Zilog Z-80 Macroassembler (with the same powerful features as all our assemblers)
- powerful linker that will link up to 128 files. Com files may start at any address
- Intel 8080 to Zilog Z-80 Source Code Converter (to convert all your Intel source to Zilog Syntax in one simple step)
- COM to Hex Converter (to convert your object files to Hex for PROM creation, etc.)
- 52 page User Manual

8086/88 Assembler with Translator $99.50

Available for MSDOS, PCDOS, or CPM/86! This fully relocatable macro-assembler will assemble and link code for MSDOS (PCDOS) AND CPM/86 on either a CPM/86 or MSDOS machine. This package also includes:

- An 8080 to 8086 source code translator (no limit on program size to translate)
- A Z-80 to 8086 translator
- 64 page user manual
- 4 linkers included:
  - –MSDOS produces .EXE file
  - –CPM/86 produces .CMD file
  - –Pure object code generation
  - –Object code and address information only

Linker features:

- Links up to 128 files
- Submit mode invocation
- Code, Data Stack and extra segments
- Handles complex overlays
- Written in assembly language for fast assemblies.

Z-8000 Cross Development Package $199.50

Instant Z-8000 Software! This package allows development and conversion of software for the Z8001, 8002, 8003 and 8004 based machines on a Z-80, Z-8000 or 8086 machine. This powerful package includes:

- a Z-80/8080 to Z-8000 Assembly Language Source Code Translator
- Z-8000 Macro Cross Assembler and Linker

The Translators provide Z-8000 source code from Intel 8080 or Zilog Z-80 source code. The Z-8000 source code used by these packages are the unique 2500AD syntax using Zilog mnemonics, designed to make the transition from Z-80 code writing to Z-8000 easy.

All 2500 AD Assemblers and Cross Assemblers support the following features:

Relocatable Code — the packages include a versatile Linker that will link up to 128 files together, or just be used for external reference resolution. Supports separate Code and Data space. The Linker allows Submit Mode or Command Invocation.

Large File Handling Capacity —the Assembler will process files as large as the disk storage device. All buffers including the symbol table buffer overflow to disk.

Powerful Macro Section— handles string comparisons during parameter substitutions. Recursion and nesting limited only by the amount of disk storage available.

Conditional Assembly—allows up to 248 levels of nesting.

Assembly Time Calculator— will perform calculations with up to 16 pending operands, using 16 or 32 Bit arithmetic (32 Bit only for 16 Bit products). The algebraic hierarchy may be changed through the use of parentheses.

Include files supported— Listing Control—allows listing of sections on the program with convenient assembly error detection overrides, along with assembly run time commands that may be used to dynamically change the listing mode during assembly.

Hex File Converter, included —for those who have special requirements, and need to generate object code in this format.

Cross reference table generated—

Plain English Error Messages—

System requirements for all programs: Z-80 CP/M 2.2 System with 54k TPA and at least a 96 column printer is recommended. Or 8086/88 256k CP/M-86 or MSDOS (PCDOS).

Cross Assembler Special Features

Z-8—User defined registers names, standard Zilog *and* Z-80 style support. Tec Hex output option.

8748—standard Intel *and* Z-80 style syntax supported.

8051—512 User defined register or addressable bit names.

6800 Family—absolute or relocatable modes, all addressing modes supported. Motorola syntax compatible. Intel Hex or S-Record format output.

6502—Standard syntax or Z-80 type syntax supported, all addressing modes supported.

8086 and Z-8000 XASM includes Source Code Translators

	Z-80 CP/M®	ZILOG SYSTEM 8000 UNIX	IBM P.C. 8086/88 MSDOS	IBM P.C. 8086/88 CP/M 86	OLIVETTI M-20 PCOS
8086/88 ASM			$ 99.50	$ 99.50	
8086/88 XASM	$199.50	$750.00			$199.50
80186 XASM *new*	199.50	750.00	199.50	199.50	199.50
32000 (all) XASM *new*	299.50	750.00	299.50	299.50	299.50
68000,08,10 XASM *new*	199.50	750.00	199.50	199.50	199.50
Z-8000™ ASM		750.00			299.50
Z-8000 XASM	199.50		199.50	199.50	
Z-80 ASM	49.50				
Z-80 XASM		500.00	99.50	99.50	99.50
Z-8 XASM	99.50	500.00	99.50	99.50	99.50
6301(CMOS) *new*	99.50	500.00	99.50	99.50	99.50
6500/11 XASM *new*	99.50	500.00	99.50	99.50	99.50
6502 XASM	99.50	500.00	99.50	99.50	99.50
65CO2(CMOS) XASM *new*	99.50	500.00	99.50	99.50	99.50
6800,2,8 XASM	99.50	500.00	99.50	99.50	99.50
6801,03 XASM	99.50	500.00	99.50	99.50	99.50
6804 XASM *new*	99.50	500.00	99.50	99.50	99.50
6805 XASM	99.50	500.00	99.50	99.50	99.50
6809 XASM	99.50	500.00	99.50	99.50	99.50
8748 XASM	99.50	500.00	99.50	99.50	99.50
8051 XASM	199.50	750.00	199.50	199.50	199.50
8080 XASM	99.50	500.00	99.50	99.50	99.50
8085 XASM	99.50	500.00	99.50	99.50	99.50
8096 XASM *new*	199.50	750.00	199.50	199.50	199.50
1802 XASM	99.50	500.00	99.50	99.50	99.50
F8/3870 XASM	99.50	500.00	99.50	99.50	99.50
COPS400 XASM	99.50	500.00	99.50	99.50	99.50
NEC7500 XASM	99.50	500.00	99.50	99.50	99.50
NSC800	99.50	500.00	99.50	99.50	99.50
Subtotal	$ _____	$ _____	$ _____	$ _____	$ _____

Name _____
Company _____
Address _____
City _____ State _____ Zip _____
Phone _____ Ext. _____
Make and model of computer system _____
☐ C.O.D. *(2500AD pays C.O.D. charges)*
☐ VISA or MasterCard #, Exp. Date (mo./yr.)
_____
Signature _____

TO ORDER. **Simply circle the product or products you want in the price columns above, enter the subtotal at the bottom of that column and add up your total order. Don't forget shipping/handling.**

Total $ _____

Check one:
☐ 8″ Single Density
☐ 5¼″ Osborne
☐ IBM P.C.
☐ Cartridge Tape
☐ Apple (Softcard)
☐ Kaypro DSDD
other formats available, please call!

shipping/handling ($9.50 per unit, U.P.S. Blue Label, $25.00 per unit for Int'l. airmail) $ _____

Total Order $ _____

CP/M is a registered trademark of Digital Research, Inc.

218

2500AD SOFTWARE INC.

P.O. Box 4957, Englewood, CO 80155, (303) 790-2588 TELEX 752659/AD

Language Learning Tools

WATCOM's language translators make learning languages and developing programs in them a pleasure.

JOHN S. MALLOZZI

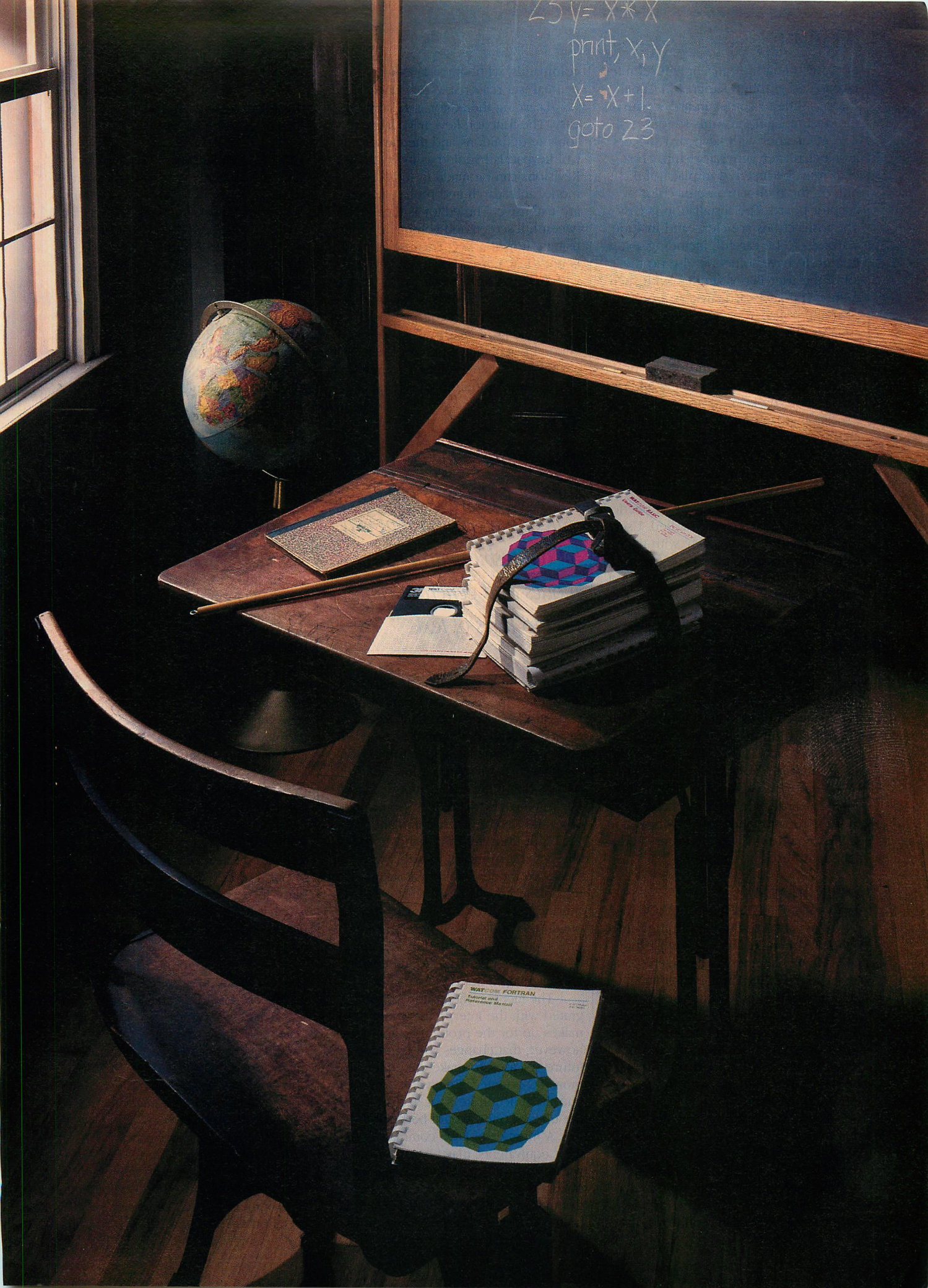
WATCOM is a company that is closely associated with the University of Waterloo, in Ontario, Canada. The university is responsible for the well-known WATFOR and WATFIV FORTRAN compilers, and the WATBOL COBOL compiler, each of which is aimed at students learning the language on a mainframe computer. WATCOM has produced a set of language translators for the IBM PC: one each for BASIC, Pascal, FORTRAN, COBOL, and APL. These translators also are aimed at a student audience; but, taking advantage of the PC's highly interactive nature, they are interpreters rather than compilers. For BASIC and APL, this is not unusual; for Pascal, FORTRAN and COBOL, program development becomes quite different.

The WATCOM languages come with an "integrated" editing facility; that is, the editor and interpreter for each language are sufficiently tied together so that the programmer may regard the combination of editing and running the program to be a single activity. This integration, together with the fact that the languages are interpreted, makes debugging a much simpler chore. Again, this is not unusual in BASIC or APL, but it is quite unusual in other languages (although it is becoming less so).

This review will look at the WATCOM languages primarily from the point of view of education, both in the sense of teaching the languages in a school setting and of an individual learning on his own. We will also look at the package from the point of view of program development, since the interpretive environment is well-suited to that phase of software writing.

The WATCOM language package comes with 10 diskettes, two for each language, and 11 manuals, two for each language and one for the editor. The manuals for a language consist of a user guide, directed to the IBM PC, and a tutorial and reference manual, which is not PC-specific. The quality of the manuals is generally quite high. The tutorials may be used in conjunction with a number of programs on the disks; they provide a somewhat curt, but reasonable introduction to the languages. Once the user is familiar with a language, the reference section becomes a valuable resource. The user guides provide information on the PC implementa-

PHOTOGRAPH • SUSANNE BUCKLER


25 Y = X * X
print, X, Y
X = X + 1.
goto 23
WATFOR FORTRAN
Tutorial and
Reference Manual

tion, including files, machine-level interfacing, and graphics support. All in all, there is little to complain about with the documentation of this system.

APL comes with keyboard stickers and a character generator chip. The chip must be inserted in place of the one on the monochrome or color adapter board. The installation is easy. The instructions are clear, and include warnings to be careful, including a suggestion to have someone else do the job if the user is not familiar with such things. The keyboard stickers are unpleasant, transparent little plastic squares, with red letters. Whether or not they are applied will depend on how often APL is used and on the user's general attitude toward stickers and symbols on the keys. More important is the fact that running APL without the character generator installed isn't really a choice, so that the user must give up some of IBM's original character set. The affected characters are "seldom used," but probably one will be needed the day after the chip is installed. Users may prefer IBM's approach, with the APL characters generated by software.

The software is not copy-protected. Each language must be installed by running a program contained on one of the diskettes. The translator being installed has a serial number, which must be typed in. Thereafter, the serial number is displayed, along with a message about the fact that copying software "is THEFT" each time the editor is loaded. The installation procedure is only a minor bother, and is certainly worthwhile since unlimited back-ups are allowed.

Memory capacity of 192KB is necessary in order to run the system, and more is recommended. Everything is in main storage, which provides great speed, but uses up memory.

ENVIRONMENT

The Pascal, FORTRAN, and COBOL interpreters are integrated with the same editor (which may also be invoked without any language interpreter); to a lesser extent, the editor is also integrated with the BASIC interpreter. From within the environment established, the user may enter or change source code, run a program, or debug a program. When an error is found, he can debug or return to the editor, with the problem line highlighted.

The editor is a full-screen editor that makes use of the function keys and permits limited reconfiguration ability: the function keys cannot be redefined, but the user can set foreground and background colors and screen mode. On-line help is available only as a list of commands, without explanations; the manual is a reasonable reference.

Learning the editor is relatively easy. The function keys have fairly reasonable definitions and the cursor control keypad is handled as in PC BASIC. Revisions can be made in full-screen mode, and search and replace facilities are extensive. The editor is toggled between full-screen and command mode using a function key. This use of modes is not ideal, particularly on a keyboard with function keys, but it works well.

Essentially, the editor has all that is needed to enter and edit programs, but is a bit old-fashioned in its modes and general orientation to lines of text. For example, to change **oldtext** to **newtext**, type **change/oldtext/newtext/**, or just

Essentially, the editor has all that is necessary to enter and edit programs, but is a bit old-fashioned in its modes and general orientation to lines of text.

c/oldtext/newtext/. This will make the change for the *first* occurrence of **oldtext** on the current line only. The command ***c*/oldtext/newtext/** changes all occurrences on all lines. A number of commands refer to line numbers, even though line numbers are not displayed. For instance, to write lines 12 through 25 to file **f.txt**, give the command **12,25put f.txt**. It is very handy to be able to save pieces of files this way, but the more modern editors allow marking and writing of blocks instead of requiring the use of line numbers. (To obtain the undisplayed number of a line, type **#** in command mode. Reference may also be made to the current line by using a dot (.).)

The fact that the editor is integrated with the interpreters more than makes up for the problems. One hopes, however, that changes will be made in future releases, to make the editor more PC-oriented. Being able to redefine the function keys, or at least define the unused ones, would be helpful.

A number of DOS commands, or their equivalents, may be given from within the editor: a directory may be obtained, or a file deleted (the latter through the command SCRATCH). It is quite possible to use DOS only to invoke the Waterloo system itself, just as for IBM PC BASIC. To print a file, type **put printer**; to save it, **put <filename>**; to read it, **get <filename>**.

INTERPRETATION

The most striking feature of these translators is the fact that they are interpreters, rather than compilers. When a language is interpreted, as BASIC usually is, the program is translated "on-the-fly," at runtime. Because of this, the program text (source code) is available at runtime, which allows a debugging tool to provide important runtime information to the programmer, such as variable names. Inclusion of such a good, high-level debugging tool is "cheap" if the language is interpreted.

The typical BASIC environment illustrates the advantages: When an error is encountered, the source text in which the error was found can be displayed immediately, execution halted, and the programmer permitted to check around, print, or reassign values of variables, change the source code, or perform other correcting functions.

When a language is compiled, as is usually the case with Pascal, FORTRAN and COBOL, the program is translated completely into object code, and only the machine language is ordinarily available when the program is executed. When a runtime error occurs, a low-level message is given, perhaps involving hexadecimal addresses and offsets, with no direct way to relate the problem to the original source program and its identifiers. Even systems, such as UCSD Pascal, which integrate the compiler with the editor, are helpful at compilation time, but not runtime. (Turbo Pascal, along with giving an error number and a hexadecimal address, also can do a search of the source code to put the programmer in the right place to correct the error; this is unusual, but it still does not permit examination of the values of variables, changing them, and so on.

Of course, there are BASIC compilers, and Pascal interpreters; these languages are not tied to a particular mode of translation. An ideal development system would have an interpreter with integrated tools and a compiler, the latter producing optimized code and used only when development was complete. The advantages of compilation are strongest in a production setting (where programs are run much more often than translated); execution is fast and main storage is used efficiently. In a learning situation, or in a development environment, the advantages of interpretation are strongest. This fact alone

Up Your AT™ for $56 per Megabyte!

■ While our specifications certainly speak for themselves, we thought you still might like to hear from some of our users:

■ "Emerald Systems expands the potential of PCs by providing the ability to access large amounts of data on line, quickly and reliably."
Terry Baptiste, Computerland, Lafayette, Ca.

■ "Service and support is great, which is an unusual experience. Emerald's software for backup and restore is invaluable. Can't put a price on it. Productivity and efficiency has increased at least 50%!"
Bruce Kittinger, Pinon Systems, Ft. Collins, Co.

■ Runs like a champ with 3-Com Ethernet."
Alvaro Ramirez, Micro Age, Miami, Fl.

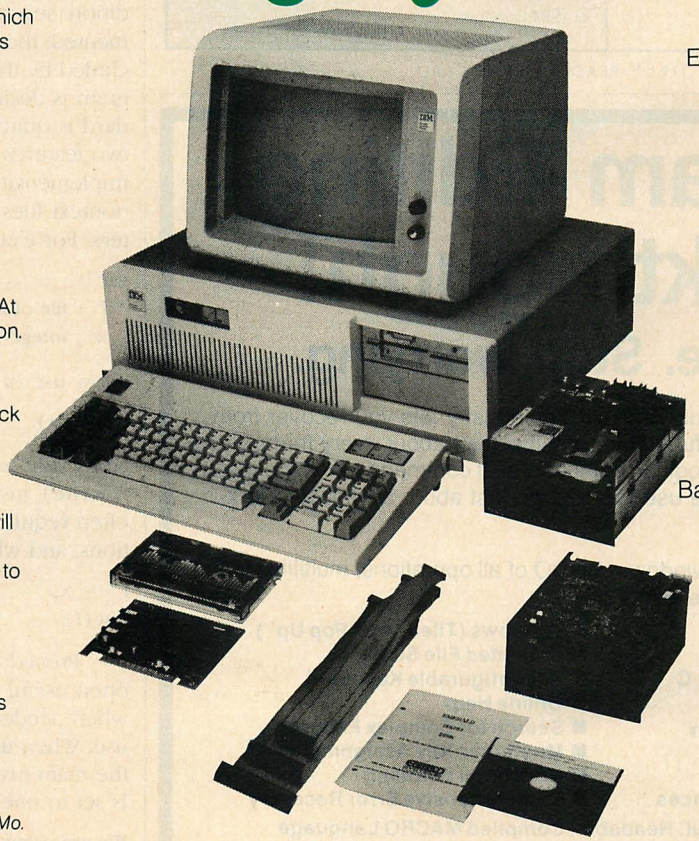
■ ". . . high capacity and flexibility. At last, a tape back up we can count on. And the price is right!"
John Acres, EDT, Las Vegas, Nv.

■ "The speed at which you can back up is very impressive."
Jim McEwen, Mercy Hospital, Portland, Me.

■ "When Emerald says your unit will be there on Thursday, it's there on Thursday! Delighted we were able to exceed the usual 32 megabyte restriction."
Steven Mayer, Take One Company, New York, N.Y.

■ "The Emerald 65 MB hard disk is extremely easy to install and work with. Emerald has a complicated piece of equipment made easy to use."
Tom Edler, Jewish Hospital, St. Louis, Mo.

■ "Our Emerald fixed disk installed quickly and easily. Emerald's reliable disk and tape backup further enhances LIBRA's high function accounting software."
Kenn White, Libra Programming, Salt Lake City, Ut.



HARD DISK
We've broken through the 32 MByte DOS barrier!

Now you can create up to 240 MByte databases on one file, with multiple volumes per disk drive. You get 140 times more memory with a 30% increase in access speed!

And that's not all:

Expands up to 280 MB for as little as $56 per megabyte.
6 expansion slots
Up to 24 volumes
Drive sizes: 40,70,140
Internal or External
Also for PC,XT, and compatibles
Supports all PC compatible networks
Simple menu-driven installation

TAPE BACKUP
¼" cartridge
½" 9 track
60 Megabytes
No lost data from tape run out
Backup and Restore Utility (BRU™) software included
LAN Compatible*

***FREE APPLICATION GUIDE**: "IBM LAN Installation and Implementation."

Call (619) 270-1994, or write to

Emerald™ Systems Corporation
Mainframe Storage for Micros
4901 Morena Blvd, San Diego, 92117

Distributed by Manchester Equipment of NYC, and selected Entre, MicroAge and Computerland stores.

EMERALD™
SYSTEMS CORPORATION
Mainframe Storage for Micros™

FIGURE 1: *Output of Listing 1*

```
The value is  2

We are using...procedure one.
The value is now  5

We are using...procedure two.
The value is now 25

All done.
```

The program in listing 1 illustrates the use of procedural parameters in WATCOM Pascal.

may account for a good part of BASIC's continued popularity: because BASIC environments are interpreted, it makes them simply easier to deal with.

PASCAL INTERPRETER

WATCOM'S implementation of Pascal is very close to ANSI standard (ISO standard is the same, except that it includes conformant array parameters, which are not present in WATCOM Pascal). In addition, several extensions are implemented; these extensions can be excluded by the user if a standard program is desired. Support for the standard is quite complete. For instance, two features often missing from Pascal implementations are **read** and **write** for nontext files and procedural parameters. For example, the declarations

```
var
  f : file of integer;
  x : integer;
```

allow use of the statement

```
write(f,x);
```

(if the file has been opened by **rewrite**), instead of the following form often required in other implementations, and which is available if desired):

```
f^ := x;
put (f);
```

Procedural parameters are not often useful, but are hard to simulate when needed. Listing 1 illustrates their use. When **use(one,x);** is executed in the main program, parameter **p** in **use** is set to **one**. Therefore, after

```
We are using..
```

is printed, the statement **p(n)** is equal to **one(n)** which completes the line with the following:

```
.procedure one.
```

and sets the value of **n** equal to **2*n+1**. Figure 1 shows complete output.

The quality of the translator is very high; the language definition is scrupulously followed. Many errors that would otherwise make programs nonstandard are found and error messages given. This is the case for unassigned variables, attempts to use a loop counter after the completion of a loop, and out-of-range variables. One case *not* dealt with is out-of-range integers; for example, the assignment (for **x** an **integer**)

```
x := maxint + 1;
```

gives no error (**x** now has the value −32768). Generally, though, runtime errors are handled beautifully.

Syntax errors are too often handled with the message **syntax error**, when the interpreter could be more specific. I should add quickly that this approach is usually preferable to an attempt to correct the error—as in the IBM compiler—which is often followed by a cascade of spurious error messages. Nevertheless, there is middle ground.

Consider listing 2, for example. The program contains three problems, marked {1}, {2}, and {3}. Problem {1} is a syntax error: there is a missing semicolon at the end of the statement. Problem {2} is the use of variable **x** before it has been defined. Problem {3} is an attempt to use a loop counter after completion of the loop. While most Pascal compilers will give a **semicolon expected**, or (in UCSD) **terminator expected**, message (and the IBM compiler will insert a semicolon to permit continuation of compilation), all WATCOM says is "Incorrect syntax near or before. . . ." followed by a listing of the line after the one with the error.

On the other hand, after correcting the syntax error, I tried this program on the Turbo, IBM, and UCSD compilers, and, after a successful compilation (there were no further syntax errors), the programs were run. No runtime errors were reported; a value was used for **x** (whatever was "lying around"—a different value in each system) and no problem was found with accessing **i** after the loop. (The value used for **i** was 5 for Turbo and 6 for both IBM and UCSD). Nothing about this is illegal, but the type of bug in {2} can be hard to trace, and the usage in {3} can be dangerous. For these, the WATCOM interpreter reports "The value of 'x' is undefined" and (after a value is given to **x** to allow the program to run) it says the same about variable **i**. The first is nice for learners and developers; the second is not only nice, but quite unusual and indicative of the care taken to follow the standard, and to report deviations from that standard.

WATCOM

Extensions to standard Pascal provided by the WATCOM implementation include variable-length character strings, random access to files, and graphics support. The first is different from the implementation used by UCSD, Turbo, and some others, but includes all necessary string-processing procedures. An advantage of the WATCOM approach is that departures from the standard are minimal. A string is declared as a packed array of characters, but its length is permitted to vary. The ambiguous single character (Is **'a'** a character or a string containing one character?) is handled by permitting the use of double quotes for strings, so that **'a'** is a character, **"a"** is a string, and **'ab'** and **"ab"** are equivalent notations for the string containing the first two letters of the alphabet. The empty string must be written " ". Names for string procedures are unique to this implementation, and somewhat unpleasant; for example, StrConcat is used to concatenate strings, and StrLen gives the length of a string.

The user may "seek" positions in a random file and append information to a sequential file. These extensions are provided in a manner very consistent with the standard. Interactive input, the bane of many implementations (UCSD and Turbo Pascal among many others), is accomplished using "lazy I/O," which allows standard Pascal, but with normal screen interaction. One interesting feature is the ability of the program to read from wherever the cursor is positioned. For example, if the operator moves the cursor (using the cursor pad) back to data already read, it will be read again. On the other hand, no means of controlling the cursor from within a program (as with GotoXY in UCSD) is provided—an inexplicable omission. Function keys can be set and read, and **peek** and **poke** are provided.

Graphics support is similar to that provided by BASICA: with the capability to draw lines, circles, and boxes, and to paint. While many might prefer Turtlegraphics, WATCOM has one of the few relatively complete graphics extensions in a Pascal implementation, and the one provided makes for a very smooth transition for BASIC programmers. Remember that the system is interpreted, so it is not really faster than BASIC.

Since WATCOM Pascal is interpreted, the environment provided is similar to that to which BASIC programmers are accustomed. The system is invoked from DOS with

```
wpas
```

This activates a batch file which loads the editor and the interpreter. To re-

turn to DOS, type **bye**. The user finds himself in the editor, whereupon the source code can be entered. When done, return to command mode (by striking the F9 key), and simply type

```
run
```

The program immediately begins to execute; if an error occurs, the user will find himself in the debugger. At this point, he may press <enter> to attempt to continue; type **q**, to quit execution and return to the editor, with the offending statement highlighted; or remain in the debugger. The response is very fast; no disk access is required.

If he elects to stay in the debugger, he may type **e** followed by a Pascal statement to execute that statement as if it were part of the program. For example, to print out the values of variables **x** and **y** at the time the program was interrupted, just type

```
e writeln(x,y);
```

This is one of the major advantages of this package. The program can also be stopped and the debugger entered by calling the procedure **pause** from within the program. Then, use **s** to singlestep through the program, stopping anywhere along the way to type out values, assign new values, etc. Type **q** or **c** to quit or continue execution.

BASIC programmers will feel very much at home in this environment (once they have learned some Pascal). Pascal programmers coming from a typical environment for that language will find that development time is shortened dramatically. Error correction is simple; no disk accesses are needed during any phase, unless the user wants to save the program or use file I/O. Somewhat disconcerting, at first, is the fact that errors in the main program are found before those in procedures, since translation begins with the declared objects (constants, variables, etc.), then goes on to the first statement executed.

FORTRAN AND COBOL

Both of these interpreters use the same editor and debugger as does the Pascal interpreter discussed above. FORTRAN is interpreted as each statement is executed, but a COBOL program has its entire syntax checked first, then it is executed. An error found during syntax analysis does not invoke the debugger; only a runtime error has this effect. Thus, the COBOL translator acts more like a compiler (but with the development and debugging speed capacity of an interpreter). The interactive debugger is available here, as it is with Pascal: typing **e statement**, where **statement** is

a statement in the language being interpreted, executes that statement as if it were included in the program.

WATCOM FORTRAN is a subset of FORTRAN 77, which also has extensions. Some unusual omissions are the DATA statement and DOUBLE PRECISION and LOGICAL variables (although single precision is 15 decimal digits, so that REAL here is DOUBLE PRECISION on many other systems). The version of FORTRAN supported is unpleasantly nonstandard. For example, it uses a different syntax for the structured IF-THEN-ELSE than does FORTRAN 77. It also has an alternative DO-loop that is more "structured" than the standard one. Since the DO loop is one of FORTRAN's few structured control statements, I question the need for an alternative, especially since the one provided is enough like the standard one to be confusing. A number of other extensions that are unique to this implementation are also provided.

One problem I see with this approach is that the ideal use of FORTRAN in an educational setting is to provide a marketable skill to those who have already learned to program in a "better" language. With this in mind, I expect the FORTRAN I use to be reasonably standard. At the very least, standard programs should work. Unfortunately, perfectly ordinary programs that start comment lines with a C will cause an error here because in WATCOM FORTRAN, comments are started with an asterisk. (The error message is confusing, too. The comment is passed over, and WATCOM says that the **DIMENSION** statement is misplaced.)

WATCOM FORTRAN includes graphics routines and other microcomputer-oriented extensions, such as function-key access. The interactive environment makes development and learning pleasant, as with the Pascal interpreter. Unlike Pascal, however, standards are not adhered to, and, as a result, its end usefulness is compromised. Another concern is that it is especially odd to deal with a version of FORTRAN that cannot handle separate compilation of subroutines. (No **include** directive exists. The user can, however, **get** the file containing the desired subroutine, which will physically insert the text. Although this does not change the permanent file, it isn't really satisfactory: the **get** has to be done each time the program is run.)

WATCOM COBOL is a dialect of COBOL 74 and, like most small computer implementations, it is missing the SORT and SEARCH verbs, as well as INDEXED files and REPORT WRITER. These are not unreasonable omissions, however, particularly for someone learning the language. The CALL statement is unsupported, except for machine-language interfacing. This translator is much more likely to be useful as a learning tool than the FORTRAN interpreter, since standard programs run more easily. In contrast to the Pascal implementation, errors in syntax are reported with informative error messages. Program development is well-supported, and using this system to teach COBOL—to oneself or to others—is a pleasure. It also should be fairly easy to switch to a mainframe implementation.

Unfortunately, perfectly ordinary FORTRAN programs that start comment lines with a C will cause an error because in WATCOM's implementation, comments are started with an asterisk.

WATCOM BASIC AND APL

The implementation of BASIC is very interesting, from the standpoints of system facilities provided and language supported. The BASIC environment is essentially that with which most programmers are familiar, though some of the commands differ; for example, SCRATCH deletes a file, and CLEAR is used to start a new program (instead of typing NEW). Facilities such as AUTO, to generate line numbers automatically, and RENUM, to renumber the lines in a program, are also provided.

It also includes some features not found in many other BASIC implementations. One of these is the TYPE command, which works very much as in DOS, with the added feature that it pauses when the screen is full. Probably the nicest feature, particularly when it is considered as an educational tool, is the LOG command. Listing 3 is a short BASIC program that has been saved in b:DEMO.BAS. From the Ready prompt (equivalent to PC BASIC's **ok**), type

```
log loglist
```

From here on, all responses from the system are logged in the named file, with the prefix ">" to indicate that it comes from the system; anything typed

by the user is logged with the prefix "<". This continues until **endlog** is typed, and it, therefore, becomes the last logged line. See figure 2.

Editing programs can be done using line numbers and retyping, as with all BASICs, but the WATCOM editor used in the other languages can also be used in BASIC. Although this editor does not work with BASIC in the sense that, for example, line numbers are automatically generated, if a program is typed in without line numbers, room is left for them. When the editor is exited (using **bye**) the system returns to BASIC, whereupon the **renum** command can be used to supply line numbers. Commands within the editor, including function key usage, are the same as for the other languages.

On the other hand, no assistance is given within BASIC itself; function keys are not available, and full-screen editing cannot be done as it can in PC BASIC. Nevertheless, editing isn't difficult. Just type **edit** to begin editing and **bye** to return to BASIC. The **run** command from within the editor both exits to BASIC and runs the program.

The WATCOM version of BASIC is quite extensive, including graphics support, PRINT USING statements, error interrupts, and all the other extras built into modern personal computer BASICs. In addition, however, this implementation contains quite a bit more; it presents the closest thing yet to the proposed new BASIC standard. Various loop structures (WHILE, UNTIL, and just plain LOOP, among others) and a nice conditional block, using IF-ELSEIF-ELSE-ENDIF, are provided. It does not have a CASE statement; instead, the GUESS statement—a rather idiosyncratic statement unlike any I have used before—is included. Some users may prefer to have the CASE structure.

WATCOM BASIC permits the definition of multiline functions, but it goes much farther. Named procedures, with local variables and parameters, are also available. This is a huge improvement over the GOSUB statement, which is just a glorified GOTO statement. An example of procedure usage is contained in listing 4. The output of a run is reproduced in figure 3. Note that the procedure is called by name, that the argument (NAME$) has a different name than the parameter (PERSON$), and that the variable N in the procedure is local: the N in the main program maintains its value independently of the N inside the procedure. The program ends after executing line 110; it does not "fall into" the procedure. Also illustrated in this example is one of WATCOM BASIC's

FIGURE 2: *Sample Log File*

```
>Ready
<old b:DEMO.BAS
>'b:DEMO.BAS' : 7 lines
>Ready
<list
>b:DEMO.BAS 84/08/01 10:35:37
>
>    10 REM THIS IS A DEMO
>    20 print "WATCOM BASIC"
>    30 print
>    40 for I = 1 to 5
>    50   print I,I^2
>    60 next I
>    70 end
>Ready
<run
>Executing...
>WATCOM BASIC
>
>  1                          1
>  2                          4
>  3                          9
>  4                          16
>  5                          25
>Ready
<endlog
```

Lines prefaced with ">" were displayed by BASIC; those that are prefaced with "<" were typed by the user.

loop statements, that is, **loop**, and the **quit** statement, which exits from the closest surrounding loop.

Thus, the WATCOM implementation of BASIC is quite strong; it makes BASIC a real programming language. The fact that it is not the same as Microsoft BASIC is a problem for those who want to develop programs for use on many machines, but, if you must use BASIC, this is good implementation.

APL is a language that either must be learned to make one a true programmer, or must be avoided, lest he become a poor programmer, depending on who is asked. It is better to avoid this controversy, and just note that, if someone wants to learn APL, the WATCOM implementation is a good place to do so. The price paid is installation of a new character generator chip, mentioned earlier.

The implementation contains no unpleasant surprises for APL lovers, and also includes some PC support. For example, the shift-PrtSc key sends a screen of text to the Graphics Printer, with the APL symbols properly printed.

FIGURE 3: *Output of Listing 4*

```
Executing...
What is your name?   John
Hi John.  Nice to meet you!
Please type a number  34
Your number is  34
Come again.
What is your name?   Mike
Hi Mike.  Nice to meet you!
Please type a number  456
Your number is  456
Come again.
What is your name?   done
 2 people visited.
Ready
```

This output of listing 4 illustrates the use of named procedures in WATCOM BASIC.

(Ctrl-PrtSc, however, does not work.) Function keys can be read and set from within functions, and graphics support is provided, the same variety that is provided in Pascal (that is, similar to that provided by IBM's BASICA). Editing can be done either with a built-in, standard APL line editor, or using the full screen, as in IBM BASICA—move the cursor to the line to be changed, make the changes, and press the enter key. System functions are plentiful, and provide, all in all, a good environment for learning or development. For someone who likes APL, or wants to learn it, and can use other languages in the package (and does not mind changing the character generator), this implementation of APL is recommended.

The WATCOM languages—Pascal, FORTRAN, COBOL, BASIC, and APL—are implemented by interpreters integrated within an interactive, development-oriented programming environment. This environment is a solid one for either learning or developing programs in the languages. While a common environment is not a serious consideration for the self-contained APL, it has important consequences for the other four languages. For example, BASIC can be learned using line editing first, then gradually making use of the general editor. The transition is easy to make because of the integration. From there, the switch to Pascal, FORTRAN or COBOL is essentially a matter of learning the languages: the environment is the same. This is very different from the usual state of affairs, where, for example, learning Pascal means a new editor, at least, and possibly even a new operating system. It is common to confuse

language details with editor features, to be uncertain about just where in the system he is, and, to be under more pressure than learning a new language should cause.

The language implementations are not of even quality, although all have strong points. Pascal is very standard, with reasonable extensions; syntax errors, however, are not reported well. COBOL has excellent error diagnostics, and is an appropriate subset of standard COBOL. FORTRAN, on the other hand, is idiosyncratic, and less useful for those who will someday move to other systems. BASIC is a very modern version of the language, quite possibly like the BASICs of the future, but, as a direct result, not so compatible with other implementations. APL is implemented reasonably well. Runtime error handling is superb for all the languages. A particular strong point is the interactive debugging in Pascal, FORTRAN, and COBOL.

The system is appropriate for instruction, or for those who want two or more of these languages in an integrated package. Particularly for Pascal and COBOL, the environment can be used for preliminary program development, before using one of the professional compiler packages available. While the editor is not ideal, it is usable, has all the necessary features, and, of course, it is integrated with the interpreters and consistent from language to language. On balance, this is a system that is easy to recommend.

One last note: WATCOM has a networking facility, whereby a mainframe or minicomputer can be used as a file server. The editor has commands for using this network. This facility, which I did not test, is probably of minor interest to individual programmers, or to schools having difficulty obtaining enough micros, let alone mainframes. However, a number of institutions already have a mainframe or mini on hand, and may find this feature to be an important consideration. I would encourage those institutions to write to WATCOM for further details.

WATCOM Languages and Editor
(Prices available from WATCOM)
WATCOM Products
415 Philip Street
Waterloo, Ontario
CANADA N2L3X2
519/886-3700

CIRCLE 457 ON READER SERVICE CARD

John S. Mallozzi has been teaching mathematics and computer science at college level for 18 years. He is currently an associate professor of computer science at Iona College in New Rochelle, New York.

101 Hands-on Micro Software Evaluations. Yours for just $19 each!

The only way to really see what a software package can do is to try it out. But software evaluations take time—and money. And even then you can't be sure you didn't overlook something big.

That's why thousands of corporate micro software buyers are turning to Data Decisions for accurate, up-to-the-minute information and independent hands-on analysis.

Data Decisions gives you the results of tests done under *consistent, controlled conditions* by a specially trained team of micro experts who tell you what each product's strengths and limitations are and how suitable it is for corporate use. They also give you all the pricing and specs.

> "Just one report saved me three nights of tinkering at a computer store on my own time. It's like having a consultant sitting right there at the end of the desk."
>
> —Bob Butler, Mechanical Engineer
> Gillette Company, Boston, MA

Until recently these evaluations were available only to subscribers to MICROCOMPUTERS, Data Decisions' 3-volume monthly updated information service. Now you can receive some of the same reports our subscribers get for only $19 each, just by filling out and returning the order form below. If you include payment with your order, you'll avoid a shipping charge.

data decisions
20 Brace Road • Cherry Hill
New Jersey 08034 • (609) 429-7100

data decisions **20 Brace Road**
Cherry Hill, New Jersey 08034 • (609) 429-7100

Spreadsheets, Finance and Planning
___ EQ Business Planning Tool
___ ER CalcIT
___ ES CalcStar
___ ET Intecalc
___ EW Multiplan
___ EX Peachcalc
___ EY Perfect Calc
___ EZ SuperCalc-3
___ FA TK!Solver
___ FB VisiCalc
___ FC Visi On Calc
___ FD VisiSchedule

Data Management
___ FE Advanced DB Master
___ FF Benchmark Data Manager
___ FG Btrieve
___ FH C.I.P.
___ FJ Dataease
___ DM DataStar
___ FL Day One DBMS
___ DJ dBASE II
___ FN dBASE III
___ FO DBM II—The Integrator
___ FP Domus Correspondence Control Manager
___ FQ Fast Facts
___ FR Friday!
___ FS Instant Recall
___ FT KeepIT
___ FW Knowledge Manager
___ FX Metafile
___ FY Nutshell
___ FZ Omnifile
___ GA Perfect Filer
___ GB Personal Pearl
___ GC pfs:File
___ GD pfs:Report
___ GE Power-Base
___ GF ProBase
___ GG R:Base Series 4000
___ DN ReportStar
___ GJ Revelation
___ DK T.I.M. IV
___ GL Under Control
___ GM Versaform XL
___ DO VisiFile

Word Processing
___ GO Benchmark
___ GP Easywriter II
___ GQ Finalword
___ GR Megawriter
___ GS Multimate
___ GT New Word
___ GW Office Writer
___ GX Palantir Word Processor
___ GY Peachtext
___ GZ Perfect Writer
___ HA pfs:Write
___ HB Select
___ HC Spellbinder
___ HD Superwriter
___ HE Textra
___ HF VisiWord
___ HG Volkswriter Deluxe
___ HH Word
___ HJ Samna Word II
___ HK Wordperfect
___ HL WordStar

Graphics
___ HM BPS Business Graphics
___ HN Chart Master
___ HO dGRAPH
___ HP Fast Graph
___ HQ Graphics Department
___ HR Graftalk
___ HS PC-Draw
___ HT Peachtree Business Graphics System
___ HW pfs:Graph
___ HX Visi On Graph

Communications
___ HY BLAST
___ HZ Crosstalk
___ JA Data Capture
___ JB IBM Asynchronous Communications Support
___ JC PC/Intercomm
___ JD PC-TALK III
___ JE Perfect Link
___ JF Smartcom II
___ JG The APPLE-IBM Connection
___ JH VisiLink

Integrated Packages
___ EJ AURA
___ JK Context MBA
___ JL ENCORE!
___ EF Framework
___ EG Goldengate
___ DL Lotus 1-2-3
___ EH PC/Focus
___ JQ SeriesOnePlus
___ EI Symphony
___ JS III E-Z Pieces

Utilities
___ JT Colortext
___ JW DESQ
___ JX Norton Utilities
___ JY Oneshot
___ JZ Prokey
___ KA Quickcode

Total Quantity Ordered ..... _____
Total Cost for Reports ($19 ea.) ..... $ _____
$2 Shipping/Handling per Report ..... $ _____
Total (Sales tax will be added on invoice where applicable) ..... $ _____

Signature _____

Name _____ Title _____

Company _____ Telephone _____ (area code)

Address _____

City _____ State _____ Zip _____

A137

CIRCLE NO. 120 ON READER SERVICE CARD

LISTING 1: PROCPARM.PAS

```
program useprocparams(input, output);
var
   x : integer;
procedure one(var n:integer);
   begin {one}
      writeln('.procedure one.');
      n := 2*n + 1
   end {one};
procedure two(var n:integer);
   begin {two}
      writeln('.procedure two.');
      n := 5*n
   end {two};
{the next procedure has procedural parameter p}
procedure use(procedure p(var m:integer); var n:integer);
   begin {use}
      writeln;
      write('We are using..');
      p(n);
      writeln('The value is now ',n:2);
   end {use};
begin {main program}
   x := 2;
   writeln('The value is ',x:2);
   use(one,x);
   use(two,x);
   writeln;
   writeln('All done.');
end.
```

LISTING 2: ERRORS.PAS

```
program errors(input, output);
var
  i, x, y : integer;
begin
   y := 2                {1}
   writeln(x+y);         {2}
   for i := 1 to 5 do
      writeln(i);
   writeln(i);           {3}
end.
```

LISTING 3: DEMO.BAS

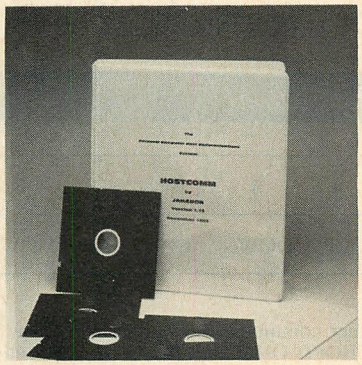
```
10 REM THIS IS A DEMO
20 print "WATCOM BASIC"
30 print
40 for I = 1 to 5
50   print I,I^2
60 next I
70 end
```

LISTING 4: NAMEPROC.BAS

```
 10 REM   AN EXAMPLE USING NAMED PROCEDURES
 20 REM
 30 N = 0
 40 loop
 50    input "What is your name?   ", NAME$
 60    if NAME$ = "done" then quit
 70    let N = N + 1
 80    call SAY_HELLO_TO (NAME$)
 90    print "Come again."
100 endloop
110 print N;"people visited."
120 REM
130 proc SAY_HELLO_TO(PERSON$),N
140   print "Hi ";PERSON$;".  Nice to meet you!"
150   input "Please type a number  ", N
160   print "Your number is ";N
170 endproc
180 end
```

Electronic Messaging

HOSTCOMM allows messages to flow from PC to PC through remote access.



AUGIE HANSEN

A Personal Computer Host Communications System, called HOSTCOMM for short, is being pressed into service for a variety of message-based applications by businesses and bulletin board operators alike. The set of programs that comprise HOSTCOMM, from Janadon, Inc./WBS and Associates, Inc., provides a means of using a PC when nobody is within striking distance of its keyboard; it activates otherwise idle PCs through remote access.

The features offered by HOSTCOMM will look familiar to anyone who has used a computer bulletin board or information service. Table 1 lists a sampling of the known public-access HOSTCOMM systems running at the end of 1984 in the United States. For many of these, the password is IBMPC, but system operators (usually called sysops) may use any passwords they wish.

The primary feature of most HOSTCOMM systems is the message facility. In its simplest form it allows messages to be left by the sysop for one or more addressees and lets users send messages to the sysop. When a user logs on to the system, any messages directed to him may be read, and responses and new messages may be sent to the sysop. The obvious extension of this is to permit any user of the system to send messages to any other user or to all users of the system or a subgroup called a special interest group. Those extensions of the basic capability may or may not be supported, depending on how the sysop sets up the system and on how much disk space is available for messages. Individual messages may be password-protected to prevent snooping.

A user may request a direct communication with the sysop using a feature called chat mode. A signal is sounded at the host end if such a request is made. It sounds until the sysop answers or until it is timed out. The timeout value is set by the sysop. Because HOSTCOMM handles only one user at a time, chat mode does not support user-to-user communications. The sysop can initiate chat mode with a user at any time, except when data loss would result from such an initiation.

Interoffice communication is an important use of HOSTCOMM. It is a fairly simple matter to set up a remote order-entry system, which allows salespersons at remote locations to phone in orders to a central facility. Also, messages can be left on line for all or designated sales personnel to help direct field activities of a dispersed organization. The same basic operations supplemented by file transfer capabilities can be used to transfer information such as periodic business reports and stock quotes. Sufficient security is built into the system to prevent casual theft of private information, but HOSTCOMM (like any other computer system) cannot be made com-

TABLE 1: *Public-access HOSTCOMM Systems*

BBS IBM HOSTCOMM

Atlanta, GA<pw=IBMPC>	404/252-4146 *[1]
Fairfax, VA<pw=IBMPC>	703/978-9592 *
Fairfax, VA<pw=IBMPC>	703/978-0921 *
Fairfax, VA	703/385-8384 *[2]
Potomac, MD	301/424-5817 *
ProMatch, Houston, TX	713/729-1257 –
Rockville, MD<pw=IBMPC>	301/949-8848 *
Springfield, VA	703/425-7229 *
Springfield, VA	703/425-6308 –
Houston, TX<pw=NFSYS>	713/890-0310 *
Toronto, Ont,<pw=IBMPC>	416/499-7023 *
Vienna, VA<pw=IBMPC>	703/522-4513 *
W. Lafayette, IN	317/463-3548 –

[1]The special symbols used in the list have the following meanings:
* mostly 24-hour, 7-day operation
– 8-12 hour nighttime activity only

[2]This system is run by Cornerstone Computer Company, which is the supplier of the HOST PCE-Mail and HOST Conference special feature programs for HOSTCOMM.

This information is based on the Public Access Message Systems list compiled by Bill Blue and distributed over USENET, The P.A.M.S. Spooler, CompuServe MAUG XA4, The Source Public 112, and PMS-Santee.

TABLE 2: *Hayes Smartmodem Switch Settings*

SWITCH #:	1	2	3	4	5	6	7	8
SETTING:	DOWN	UP	DOWN	UP	DOWN	UP	UP	DOWN

These settings may differ from those used for other communications programs. It may be necessary to change switch settings when using HOSTCOMM after they have been set up for another program.

pletely impervious to a determined and knowledgeable electronic thief.

To those in the know (special access password, etc.), HOSTCOMM provides remote program execution capabilities. Any authorized user may run special feature programs, optional HOSTCOMM modules that are tailored to meet users' special communications needs. Among the special feature programs are an order entry module (ORDTRANS), an electronic mail system (HOSTMAIL), a conference system (HOSTCONF), and a binary file transfer module (HOSTXMOD), to name a few. Standard text files may be sent and received under the basic HOSTCOMM system if the user's system has secondary storage (disk or tape), but transfers of binary files or those that contain eight-bit data require the extra-cost option that implements the XMODEM file transfer protocol.

THE SYSTEM OPERATOR'S VIEW

To run HOSTCOMM, a system must meet the following minimum requirements:

- PC-DOS 1.1 and later versions
- IBM PC or workalike with 128KB of available memory
- at least one serial port
- at least one 320KB diskette drive (two preferred)
- 80-column display and adapter
- IBM printer or equivalent (optional)
- Hayes 300- or 1,200-baud modem (the Hayes internal modem does not work with HOSTCOMM)
- an RS-232 DTE-to-DCE cable (pin 22, Ring Indicator, must be connected)

The HOSTCOMM manual gives a detailed start-up procedure, and an automatic start-up program resides on disk to help first-time users get the system running with a minimum of fuss. A few hardware and software operations must be performed before HOSTCOMM will run. The Hayes modem switch settings must conform to the settings that HOSTCOMM expects (table 2), and a few system parameters must be set in a file named SETUP.DAT. The INSTALL.EXE program provided on the distribution

diskette must be used to create and modify the set-up file. INSTALL displays a series of instructional and work screens that collect configuration and operational data from the sysop.

The instructions simply describe what each area on the installation screen is used for and how the sysop switches between areas, moves about in the areas, etc. The sysop's installation screen is the data entry screen used to form the contents of the SETUP.DAT file. Cursor pad keys select work areas. In the area at the top left part of the screen, the system logo, communications parameters, drive selections, and other data entries are made. The area at the bottom of the screen is used to identify and enable special feature programs. The data in the area at the top right of the screen control system access. The set-up session is ended by pressing F10, which writes the data to the set-up file and terminates the installation program.

The system operator's menu (photo 1) lists the available host functions. In addition to chatting with a user, the sysop can view messages, set options while HOSTCOMM is running, view files on disk, view upload/download directories, and use the mail feature. Other menu choices invoke system management functions such as viewing passwords and access level, controlling peripheral devices (printer), and stopping and restarting the program.

THE USER'S VIEW

The system requirements for the calling, or user, end of link are easily met. Any terminal or a computer running a simple communications program will suffice. Secondary storage is needed for file upload and download operations, which implies a computer rather than a stand-alone terminal. HOSTCOMM assumes it is communicating with a standard teletype-style terminal and makes no attempt to control the screen of CRT-equipped systems beyond the normal printing characters and simple format effectors such as new line, carriage return, bell, and so forth.

The first user activity is logging on to a HOSTCOMM system. Some terminals and computer communications programs require line feeds, some don't, and still others give users a choice. HOSTCOMM offers an option as its first transaction with a user logging on to the system. While getting this information, HOSTCOMM is also automatically finding out what the user's parity and data bit settings are, starting with the assumption of eight bits, no parity, and adjusting to the actual user values as

necessary. The user is also queried about the need for nulls for padding operations, such as carriage returns, that take a finite amount of time for mechanical printers.

If all goes well through the initial exchange, the user is greeted with a system logo or banner and some preliminary system-related information. At this point, the user is asked for first and last names and a location entry. Next, HOSTCOMM asks for a password, which is often either a null entry or IBMPC for public systems. Null may be entered by pressing the Return (or Enter) key on most systems. Incorrect entries are treated as a null entry, too. It should be obvious that private systems should use carefully selected passwords to prevent unauthorized access.

Several levels of log-on messages are presented to the successfully logged-on user. The first is an indication of whether any personal mail has been received for the user that remains unread. The user may either read or defer reading the mail at this point. System bulletins and general mail are presented for reading next, followed by a message stating what access level the user has. The access level specifies which menu items the user may select (1-4, 1-7, or all items, including special feature programs) and is set by the sysop on a per-user basis.

User selections are made by typing the number corresponding to the task, followed by a Return. The menu has two primary forms, depending on the user mode setting. The initial menu looks like the one shown in photo 2. The second entry allows the user to switch to the expert user mode, which displays a briefer menu (photo 3). In the expert menu, the second entry switches back to the novice user mode.

User access level three permits access to the first four selections: quit, change user mode, enter a message to the sysop, and page the sysop for a chat. Access level two allows the user access to the upload and download and change default download directory selections. Access to special feature programs, if any are installed, is granted only to access level one users.

The sysop directly controls the mailbox system, a component of the base HOSTCOMM program that permits message traffic to and from the sysop. Users cannot send directly to one another. The sysop can send messages to one or more users and a user may leave messages for the sysop. This limitation on user-to-user message traffic may seem unnecessarily restrictive, but it does have a purpose, particularly in a

distributed business environment. Like a police department or a taxi service, headquarters may want to funnel all messages through a central point in order to retain a measure of control over what is happening out in the field.

Two special feature programs get around the mail message limitation if they are installed. The first is a full-blown electronic mail system called HOST PCE-Mail, and the second is a set of electronic roundtables for discussion groups called HOST Conferencing.

If the sysop wants to support user-to-user messages, HOST PCE-Mail does the job. User numbers and passwords offer a degree of message privacy, allowing messages to be addressed to an individual or to the entire population of users (called "@"). Users may read messages addressed to themselves, but not those sent to other users. The sysop may read any message on the system. Mail items are prepared using a built-in line editor, which allows up to 25 lines of text to be written. Errors in a line are repaired by retyping the entire line—clumsy, but it works in most cases. As users read their mail items, the messages are removed from storage, freeing space for new messages. The PCE-Mail module is essential to any viable bulletin board system.

This limitation on user-to-user traffic may seem unnecessarily restrictive, but it does have a purpose.

A nice-to-have-but-not-necessary feature for bulletin boards is HOST Conferencing. The module permits one user at a time to access messages collected together under a conference name. The user may read existing messages and contribute new messages. Although this is not a real-time conference, it has its uses. The sysop creates, monitors, reorganizes, and deletes conferences at will. Together with PCE-Mail, conferencing gives system users several ways to experiment with the computer communications medium.

Another program, HOSTCCCM, permits PCE-Mail and Conferencing to share one menu entry out of the three available to special feature programs (8-10). And yet another, HOSTXTND, lets the sysop expand the special feature programs list on the main menu beyond the default of three.

We asked:

"How do you like using DATA DECISIONS MICROCOMPUTERS?"

They said:

"I looked at two other micro services but yours was far superior. Your people do a wonderful job on the hands-on evaluations. I particularly appreciate the way you look at a product from a corporate standpoint, not just from the technical point of view."

Gerald Nadeau, Systems Analyst
Analog Devices, Burlington, MA

"We act as a data resource information center for the rest of the company so we try to stay abreast of just about everything. We use Data Decisions services to answer specific questions on particular products and market sectors and find them a valuable research tool."

Stephen Lofgren, Group Manager
NEC Systems Laboratory, Boxborough, MA

"Just one report saved me three nights of tinkering at a computer store on my own time. It's like having a consultant sitting right there at the end of the desk."

Bob Butler, Mechanical Engineer
Gillette Company, Boston, MA

"I'm thoroughly sold on your services and use them almost every day. When I was interviewing with Intel, one of the first questions I asked was whether they had Data Decisions."

Virginia Franklin, Competitive Market Analyst
Intel Corporation, Austin, TX

"*Your analysts explain highly technical concepts without using jargon and yet they don't insult my intelligence. I'd been looking for this type of literature for 11 years and recommend Data Decisions to anyone who needs to know what's happening in the world of information processing.*"

Paula Brinson, Computer Manager
Hampton Roads Sanitation District, Virginia Beach, VA

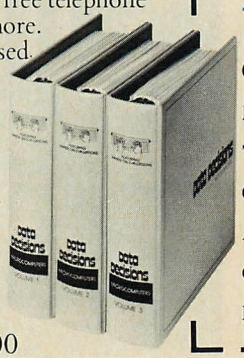
"Your telephone consultants are just outstanding—a really great group. They provide a timely, personal response to virtually any type of inquiry."

Stan Leu, Director of Marketing
DS Agency, Inc., Lubbock, TX

What are all these people so enthusiastic about? Data Decisions MICROCOMPUTERS is a 3-volume monthly updated reference service that gives you hands-on evaluations of micro software done by specially trained experts who test these products with the corporate micro buyer in mind.

You also get hardware and peripheral analyses that highlight product strengths and limitations... broad-based surveys that quickly lead you to the products that meet your needs... valid user ratings... lively monthly newsletters... action-oriented technology updates... a comprehensive vendor directory... free telephone consulting... and more.

If you're impressed by what MICROCOMPUTERS is doing for others, simply fill in the coupon to see what it can do for you. Or better yet, phone 609/429-7100.



data decisions microcomputers
20 Brace Road • Cherry Hill, NJ 08034 • 609/429-7100

data decisions microcomputers
20 Brace Road • Cherry Hill, NJ 08034

YES! Please tell me more about Data Decisions MICROCOMPUTERS.

Name_____
Title_____
Company_____
Address_____
City_____ State_____ Zip_____
Phone_____

A132/Y4

CIRCLE NO. 126 ON READER SERVICE CARD

HOSTCOMM

A useful set of programs is packaged in the HOSTCOMM utilities distribution. The programs include download rearrangement, a special download menu facility, user activity statistics, a fixed-disk facility for download files, selective back-up for fixed-disk files, and a floppy-disk facility for download directories.

Custom programs may be added to HOSTCOMM using a special feature program shell, which is BASIC code that must be used as directed to make all the necessary logical connections to access such add-on programs from the user and sysop menus.

HOSTCOMM uses HOSTRMTE to execute special feature programs written for the interpreter in a subordinate instance of the BASIC interpreter, set up with its own communications buffer and its own working environment.

Both compiled and interpreted BASIC programs may be used with HOSTCOMM. Compiled BASIC programs may be run simply by calling their names, but interpreted BASIC programs require some initial set-up. HOSTCOMM uses the program HOSTRMTE to execute special feature programs written for the interpreter in a subordinate instance of the BASIC interpreter, set up with its own communications buffer and its own working environment.

HOSTCOMM includes three utility programs that modify WordStar document files so that they may be transferred between systems. Files produced by many word processors contain control characters and characters with the eighth bit turned on. WordStar, for example, also contains lines that are sometimes too long to be handled by some systems. This is because it uses soft Returns within paragraphs to mark the ends of lines. A hard Return signals the end of a paragraph as well as its last line. Very long lines may have hard Returns placed in them by some communications programs during transmission.

HOSTCOMM's WDSTROUT and WDSTRIN programs are used in tandem on the sending and receiving sides of a link to

PHOTO 1: *The System Operator's Menu*

```
* * * The System Operator's Menu  * * *

System Operation (Wait for Call) Type  1
Restart the System               Type  2
Forced Chat with User            Type  3
Assign a Temp Access Level       Type  4
View messages to the Sysop       Type  5
View the Download Directory      Type  6
View a file                      Type  7
The Mailbox                      Type  8
To Turn Paging Bell OFF          Type  9
To Turn Lineprinter ON           Type 10
List Passwords, Access, & Dir    Type 11
Return to DOS (files closed)     Type 12


What's Your Pleasure?
```

A look at the system operator's menu gives an indication of HOSTCOMM's extensive communications capabilities.

PHOTO 2: *The Novice User's Menu*

```
You have been granted ACCESS Level  2.
Your ACCESS level allows you to use
Menu Items 1 thru 7.

Press <ENTER> when ready to proceed.

?

Your Download Directory is 3
Time Limit Remaining:  59 minute(s)

    * * * M A I N   M E N U * * *

Quit (end session)               Type 1
eXpert user mode (short menu)    Type 2
Enter text message (to sysop)    Type 3
Page the sysop                   Type 4
Upload a file                    Type 5
Download a file                  Type 6
Change download directory #      Type 7

What's Your Pleasure, WILL?  (or <H>elp) _
```

The initial user menu presents the HOSTCOMM facilities in verbose mode. The second entry allows the user to switch to the expert user menu, shown below in photo 3.

PHOTO 3: *The Expert User's Menu*

```
    * * * M A I N   M E N U * * *

Quit (end session)               Type 1
eXpert user mode (short menu)    Type 2
Enter text message (to sysop)    Type 3
Page the sysop                   Type 4
Upload a file                    Type 5
Download a file                  Type 6
Change download directory #      Type 7

What's Your Pleasure, WILL?  (or <H>elp) 2
?
?
Dnld Dir # 3
Time Left:  57 min(s).
1-Quit 2-Novice 3-Enter Msg 4-Page
5-Upload 6-Download 7-Chg dnld dir# <H>? _
```

This shortened version of the novice menu is meant for the veteran HOSTCOMM user. Entry two on the expert menu allows the user to go back to the novice version.

modify document files into a form suitable for transmission and to restore them to document format. WDSTROUT does the output filtering. It prepares files for sending by changing each soft Return to a special mark followed by a hard Return. WSDTRIN does the input filtering. It reconstructs files after it receives them by reversing the process that is used by WDSTROUT.

For conversion of WordStar files to straight ASCII text files, the WDSTRCVT program filters a WordStar file, removing any special characters to produce a WordStar nondocument file. This process removes any special formatting and attribute information from the file, and the original file cannot be reconstructed from the resulting ASCII text file.

The HOSTCOMM manual is not the prettiest one around, but it is functional. It suffers from numerous misspellings (bulliten instead of bulletin, for example) that detract from the document's readability somewhat. The reader must jump around quite a bit to follow some topics because their coverage is split between the main text and the appendices. Installation procedures are complete and clearly written, so getting HOSTCOMM up and running should pose no real problem for would-be system operators. The manual has a good glossary and an accurate index.

Details about the use of HOSTCOMM is a good paradigm for the operation of many current information services and bulletin boards. Many people are terrified of trying to log on to such services for fear of breaking something or looking dumb. Don't worry about it. Just dial a local system and have at it. Nothing will blow up; the worst that can happen is an unwanted disconnect.

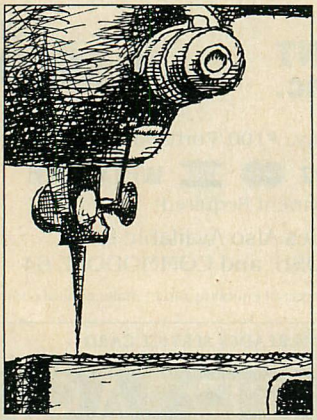
Host systems tend to be similar in the range and scope of services provided. However, HOSTCOMM has been tailored to a business environment, and it provides add-on features not typically found on many host systems intended primarily for bulletin board use.

HOSTCOMM, Version 1.13: $170
HOST PCE-Mail: $80
HOST Conferencing System and Utilities III: $125
Janadon, Inc./WBS and Associates, Inc.
7535 Little River Turnpike
Suite 300
Annandale, VA 22003
703/941-0270
CIRCLE 450 ON READER SERVICE CARD

Augie Hansen owns Omniware, a Denver-based software development company. He is a contributing editor and writes frequently about communications products.

Customized Boots

Replacing the standard DOS boot routine with the user's own program



MICHAEL ABRASH AND DAN ILLOWSKY

Booting a disk. You do it several times a day, but you have probably never stopped to wonder just how it works. The PC's basic booting procedure is actually remarkably simple, and you can, if you wish, easily replace the standard DOS boot routine on a floppy disk with your own program. This lets you make bootable disks that can be distributed freely, without having to worry about infringing on IBM's PC-DOS copyrights. Custom boot routines can also speed the booting process, save space on the disk, and permit faster disk production.

What follows is an explanation of how the PC boots and how the standard boot routine can be replaced. A sample replacement boot routine for a floppy disk is presented. The routine is functional and can be used to boot the user's own programs.

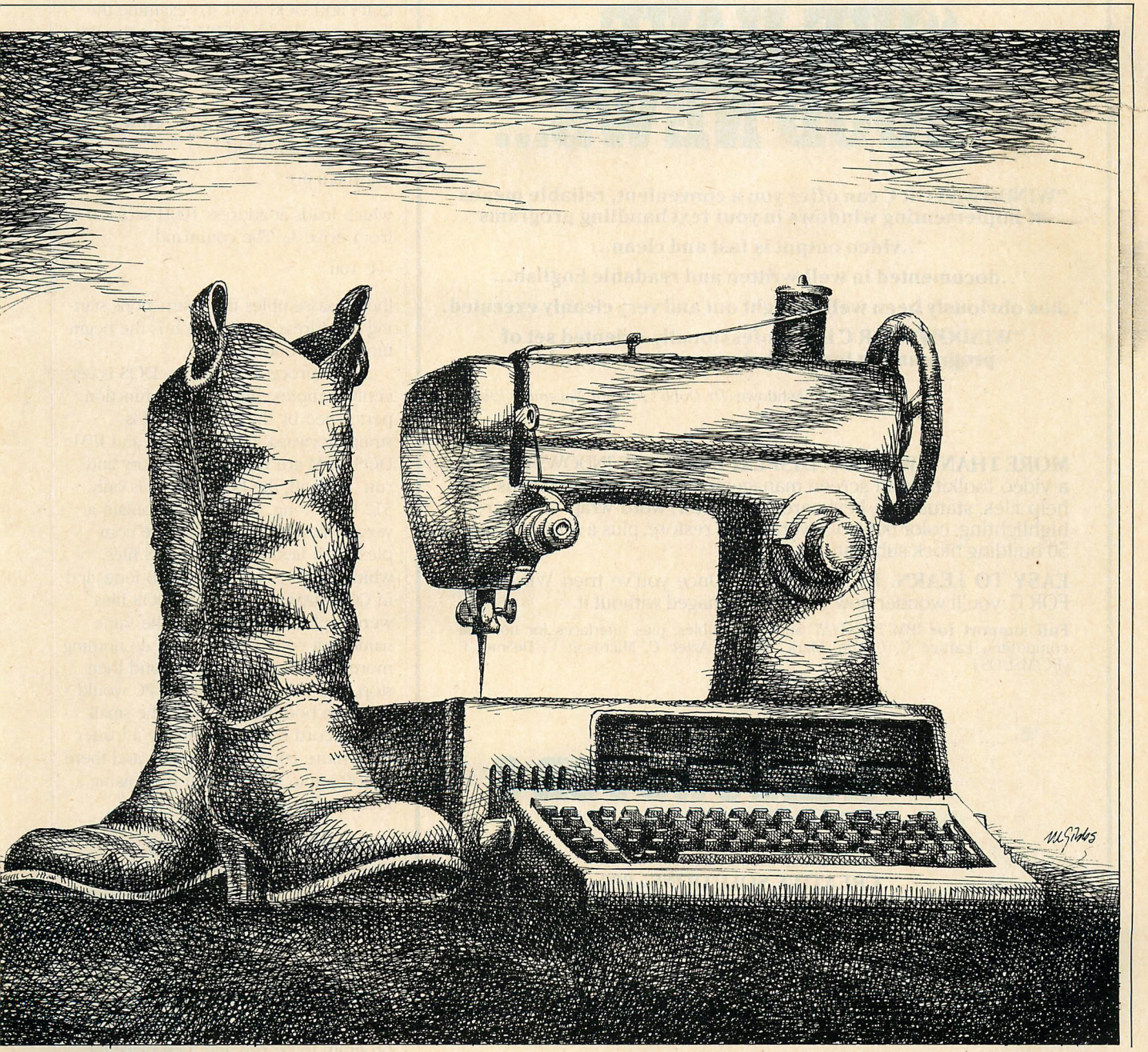
The boot record program does nothing more than load another program into memory. Normally, the program loaded is PC-DOS. DOS sees to all the details of booting, but, as you will see shortly, you can set up your own boot record to load programs into memory, bypassing DOS entirely. With the boot record program, programs start very quickly, saving the 40KB of disk space that DOS ordinarily fills. Because DOS is not necessary, a commercially distributed program need not require the user to transfer DOS to the program disk. (DOS is proprietary to IBM; it cannot be resold or distributed without a license.) A customized boot track is also useful in copy protection schemes. Finally, since it is necessary to format only those tracks that will be needed and not every track on a non-DOS disk, many hours and dollars can be saved when producing thousands of disks commercially.

All in all, a customized boot record, dedicated to loading a specific application, is fast, compact, and infringes on no copyrights; however, there is an obvious disadvantage to custom booting. DOS is not available when you use your own boot record. For many applications, this can be a major disadvantage, since DOS provides very powerful functions for input and output, particularly where disk access is concerned. However, because the BIOS functions, which are stored in ROM, are always available, programs with limited or clearly defined input/output demands can use the more primitive BIOS functions instead of the DOS functions and so can manage without DOS.

Booting, or bootstrapping, is the process of loading an operating system into memory from a disk and then initializing it. The name is derived from the nature of the process itself: an operating system being booted first loads in part of itself, which then loads in more of itself, and so on. This is not unlike the proverbial process of pulling oneself up by one's bootstraps.

When the PC is turned on, it first tests the installed memory and peripherals. Then, if a disk drive is present, the PC tries to boot from it by trying to read sector 1 from track 0, side 0 of the disk in drive A:. This sector, known as the boot record, must be 512 bytes in size. If the boot record is found, its

ILLUSTRATION • MICHAEL GIBBS



contents are loaded into memory at address 0000:70H, and a far jump is performed to 0000:7C00H to execute the routine. In short, the routine on the first sector of the disk in drive A: is loaded into memory at 0000:7C00H and run. What happens after that is up to the routine stored on that sector. (With the proper ROM, fixed-disk systems, for example the XT, can load the boot record from sector 1, head 0 of the first cylinder of the DOS partition. However, only floppy-disk boot records will be discussed in this article.)

When the DOS boot record is loaded and run, it tries to load in the rest of DOS. It does this by checking to see that the two files IBMBIO.COM and IBMDOS.COM are the first two files in the directory, and then, if they are present, loading them into memory and transferring control to them. IBMBIO.COM, which contains the interface between the high-level DOS functions and the low-level BIOS functions stored in ROM, is initialized first. When this file is loaded and given control, it checks and initializes the equipment in the PC and sets some of the interrupt vectors. IBMDOS.COM, which contains DOS itself, is then relocated to use the space that the IBMBIO.COM initialization routines used.

Once moved, IBMDOS.COM initializes DOS, sets up the DOS interrupts, and sets up a header for the command processor. (The initialization code that does this is later overwritten by the command processor and used for data storage; as you can see, DOS never lets free space go to waste.) Finally, IBMBIO.COM is invoked to load in COMMAND.COM, the command processor, which provides the A> prompt and gives DOS an interface with the user.

If you are curious, you can use the DEBUG debugger that comes with DOS to disassemble the boot record program and DOS itself. To examine the boot record with DEBUG, start the debugger with the command

```
A>DEBUG
```

and then load the boot record with the debugger command

```
—L 100 0 0 1
```

which loads at address 100H sector 0 from drive A:. The command

```
—U 100
```

then disassembles the instructions starting at address 100H, which is the beginning of the boot routine.

The process of booting DOS is certainly complex, but the basic function performed by the boot record is straightforward. IBMBIO.COM and IBMDOS.COM are read into memory and run. After all, the boot record is only 512 bytes long, so it cannot contain a very complicated program; the complexity all lies in the two DOS files, which together are over 20KB long, and in COMMAND.COM. If the DOS files were replaced with files of the same name that contained code to do nothing more than display "HELLO" and then stop, that is exactly what the PC would do upon booting that disk. The small boot record routine is merely a loader for a more complex program, and there is nothing to prevent you from using a loader tailored to your needs.

A SAMPLE BOOT RECORD

The following example of a customized boot record will load a program from consecutive tracks on the disk and start it running. This is simple enough, although getting the boot record and the program on the disk is a bit more difficult. To make a bootable disk, first create the boot record program and the program to be run and then place the

PHOTO 1: *Sample run of MAKEBOOT*

```
B>MAKEBOOT

File containing boot routine: BOOT.COM
File containing application program: SAMPLE.COM

Insert a blank disk in drive A:.
This disk will be wiped out, so be careful!
Strike any key to begin...L

The disk in drive A: is ready to boot
```

This procedure causes the program in drive A: automatically to boot the applications program SAMPLE without DOS. The user types commands shown in upper case.

boot routine on sector 1, track 0, side 0 of the disk in drive A: to be loaded and run when the PC is started. Finally, put the program on the consecutive tracks where the boot routine will look for the program when the disk is booted. The boot routine will then read the program back into memory.

Before beginning, a few points about the boot record should be made. First, make sure that a blank disk is in drive A: whenever the program that creates the bootable disk is run. After the program is run, the disk in drive A: will be useless for any purpose other than running the sample boot routine. Along the same lines, never run the program that makes a disk bootable when a hard disk is drive A:. Presumably, no one wants to overwrite any portion of a fixed disk.

Second, use .COM files and only .COM files with the sample boot routine. They can run anywhere in memory as long as all the segments are set to point to the same location, with the program starting at offset 100H in the code segment. .EXE files, on the other hand, must be corrected for their location in memory. DOS normally performs this correction automatically, but, it is wise to remember that DOS is not present now.

Third, when the boot routine is loaded, only the code segment/instruction pointer pair is well defined, pointing to 0000:7C0H. On our PCs, the data segment register contains 40H, the stack segment register contains 30H, and the extra segment register contains 0 when the boot routine is started; but there is no assurance this will be true on all PCs. Because it is never a good idea to rely on undocumented features, this boot routine is completely independent of the settings of the segment registers.

Fourth, our boot routine will load a program from eight-sector tracks, and from one side of the disk only. Since there are 40 tracks on each side of a disk (one of which is used by the boot routine), and since each of the eight sectors on a track holds 512 bytes, the boot routine can load an applications program of up to 156KB in size.

The assembly language source code for the customized disk boot routine is shown in listing 1. This routine works in the area of memory from 0020:0000H to 0020:01FFH. The 512 bytes in this address range make up the second half of the interrupt vector table, but these interrupt vectors are not used by the BIOS, so this area of memory is a safe area for us to use. (It is not, however, an "officially" safe area, since IBM has not yet documented that it is free for use.)

The first task the boot routine does is move itself from the memory location at which it is initially loaded to the free area located at 0020:0000H. This is necessary because the boot routine loads at 0000:7C00H, and a large applications program could overwrite the boot routine as the boot routine loads the program in.

Once the boot routine finishes moving itself, it reads the applications program into memory one track at a time, starting with track 1. (Track 0 contains the boot routine.) The number of tracks read into memory is determined by the value of HIGHEST__TRACK; since our applications program is less than 100 bytes long and since each track holds 4,096 bytes, we need to read only one track. The program is loaded starting at 0040:0100H. The 100H offset is necessary because all .COM programs must start at this offset from the program segment.

When .COM programs are run under DOS, the 256 bytes from offset 0 to offset FF are referred to as the Program Segment Prefix and contain useful information. Programs loaded by our boot routine will have none of this information available, since DOS will not be running; consequently, programs loaded with the boot routine should never access the 256 bytes starting at offset 0 as if that area were the Program Segment Prefix. Another important reason to steer clear of the memory from 0040:0000H to 0040:00FFH (offset 0 to offset FF in the program's code segment) is that this is the area that is used by the BIOS for the storage of its own variables.

Of course, this is a program that could just as easily be loaded into memory at any address, provided that it is higher than 0040:0100H, just as long as the offset is 100H. For example, 0050:0100H would be a perfectly valid load address; to effect this change, the user must change the "AT" clause in the definition of PROGRAM __ SEGMENT from 40H to 50H.

The boot routine shown in listing 1 uses only one side of the disk, and only eight sectors per track. To use both sides of the disk, the disk access routines and format block must be altered. To use nine sectors per track, the number of sectors to format and write, the number of format block groups, and the disk parameter block must all be changed. See Appendix A of IBM's *Technical Reference* manual for information on the disk parameter block and on disk access through the BIOS.

Finally, when the applications program has been loaded into memory, the stack segment is set to match the other three segments, as is required for a .COM program. The stack pointer value is set to 0FC00H, which will leave the maximum amount of room free for the applications program on a system that has 64KB of memory installed. In a system with more than 64KB, the stack pointer can be set to 0000H to free up the maximum amount of memory a .COM program can handle; in systems that have less than 64KB, the stack pointer value will have to be reduced accordingly. When the stack is set, a far jump is performed to 0040:0100H to begin execution of the applications program, which then runs normally.

The boot routine is assembled, linked, and made into a .COM file. Do not worry about the warning concerning the missing stack segment; the stack segment for .COM programs is set when the program is loaded to be run, and we will do this ourselves in the boot routine. The assembly language source code for the sample applications program is shown in listing 2, a program that simply confirms its existence and then enters an infinite loop.

The sample program uses no DOS functions, since DOS will not be available. BIOS function 10H, which performs screen output, is used when it is necessary to display characters.

The sample program is assembled, linked, and made into a .COM file; as with the boot routine, ignore the warning about the missing stack segment.

DISK PREPARATION PROGRAM

The program shown in listing 3 prepares a disk to be booted. First, it formats the disk to be made bootable, and then it places the boot routine and the sample program on the disk, with the boot routine placed on sector 1 of track 0 and with the applications program on all eight sectors of each track from track 1 out as needed. Our sample program is well under the 4,096-byte capacity of a single track, so we need to use only track 1.

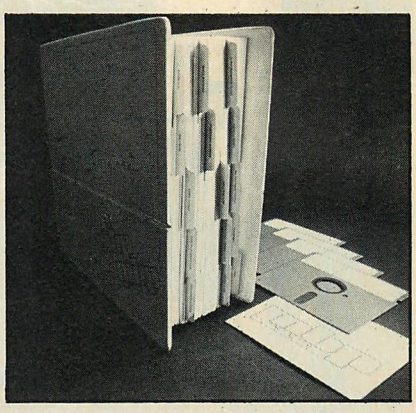
For your IBM/PC

mbp COBOL: 4 times faster, and now with SORT & CHAIN.

$750.

mbp COBOL can be summed up in one word: fast.

Because it generates native machine language object code, the mbp COBOL Compiler executes IBM/PC* programs *at least* 4 times faster (see chart). Fast also describes our **new SORT**, which can sort four-thousand 128-byte records in less than 30 seconds. A callable subroutine or stand-alone, 9 SORT control fields can be specified. And our **new CHAIN** is both fast and secure, conveniently transferring control from one program to another, passing 255 parameters. Plus, **new extensions** to ACCEPT & DISPLAY verbs give better, faster interactive programming.



GIBSON MIX Benchmark Results
Calculated S-Profile
(Representative COBOL statement mix)
Execution time ratio

mbp COBOL	Level II** COBOL	R-M*** COBOL	Microsoft**** COBOL
1.00	4.08	5.98	6.18

128K system with hard disk required. *IBM/PC is an IBM TM; **Level II is a Micro Focus TM; ***A Ryan-McFarland TM; ****A Microsoft TM.

The complete COBOL.

An Interactive Symbolic Debug Package included standard; Multi-Keyed ISAM Structure; listing options allow source & object code, map & cross-reference checking; GSA Certification to ANSI '74 Level II; mbp has it all.

It's no surprise companies like Bechtel, Chase, Citicorp, Connecticut Mutual, and Sikorsky choose mbp COBOL; make it your choice, too. mbp is available at Vanpak Software Centers, or direct; just send the coupon, or call for complete information—today.

mbp COBOL: 4 times faster.

CIRCLE NO. 165 ON READER SERVICE CARD

mbp®

mbp Software & Systems Technology, Inc.
7700 Edgewater Drive, Suite 360, Oakland, CA 94621
Phone 415/632-1555

Please send complete mbp COBOL information to:

NAME _____

COMPANY _____

ADDRESS _____

CITY/STATE/ZIP _____

PHONE _____

To handle an applications program that is more than 4,096 bytes in length, just change the value of PROG__LEN to the number of bytes needed, as shown in the directory entry for the applications program to be booted. The maximum value of PROG__LEN is a little less than 64KB, because the applications program must fit in the same segment with the boot routine and assorted messages, and because 100 bytes are lost at the start of a .COM program.

The disk preparation program is assembled and linked with the appropriate commands. The uppercase commands shown in photo 1 prepare a disk to boot with our boot routine and sample program. A boot routine and a sample program (both in .COM form) must always be present in order for the disk preparation program to run properly. On systems that are not IBM-compatible at the BIOS level, it is probably best not to use our boot routine and disk preparation program unless you know what you are doing.

After producing a bootable disk, test it by pressing Ctrl-Alt-Del to warm-boot the PC. The following message should appear on the screen:

Congratulations!
You have booted a sample applications program.

If not, repeat the steps carefully.

When you boot the disk with the customized boot routine, contrast the boot speed of this disk with that of DOS. It will become apparent that the customized routine is considerably faster. If you experiment, you will also find that you cannot stop the boot process with Ctrl-Break; because the "^C" break function is provided by DOS, Ctrl-Break will have no effect on a program that uses only the BIOS unless the vector for interrupt 1BH points to your own break-handling routine.

Any .COM program can be used with the boot routine shown in listing 1, as long as it does not use any DOS functions or interrupts. The BIOS functions, which can be used at any time, are accessed with interrupts 0 through 1FH. These provide low-level access to the display, keyboard, and disk drive; while they are not as convenient as DOS functions, the BIOS functions are adequate and always available.

Two changes need to be made to accommodate other applications programs. First, the highest track to be written by the disk preparation program and read by the boot routine should be changed to reflect the number of 4,096-byte tracks taken by the applications program. Second, the size of the buffer that holds the applications program in the disk preparation program must be adjusted. This will mean changing PROG__LEN to equal the size of the applications program. (You can read the size of the applications program from its entry in the disk directory listing.)

Two important references that discuss booting the PC are IBM's *Technical Reference* manual and *DOS Technical Reference* manual. IBM's *Technical Reference* manual contains the listing of the ROM routine that loads the boot record into memory, and lists the complete code of all the functions provided by the BIOS. IBM's DOS manual discusses the process whereby DOS bootstraps itself into memory. (The DOS 2.1 manual has been split into two volumes, with much of the technical information in the *DOS Technical Reference* manual.)

Michael Abrash and Dan Illowsky are officers of Funtastic, Inc., a software publishing house near Philadelphia. They are authors of Snack Attack, Country Fair, Cosmic Crusader, *and* Big Top *for the IBM PC and Apple II.*

LISTING 1: BOOT.ASM

```
;
;A customized boot routine, which will load an application program
;  from tracks 1 through HIGHEST_TRACK and start it running.
;
;The segment to which the boot routine will be relocated so that it
;will  be out of the way of the program to be loaded.
moved_segment       segment at 0020h
   org     0
moved_routine       label   far
moved_segment       ends
;
;The segment in which the application program will be loaded
program_segment     segment at 0040h
   org     100h
start_pointlabel    far     ;Since it is a COM program, the
                            ; application program will start at
                            ; offset 100 (hex)
program_segment     ends
;
highest_track       equ     1       ;The highest-numbered track to be
                            ;read in
;
cseg        segment
   assume  cs:cseg
;
;While this is a COM program, it is designed to be independent of its
;  location in memory, and so it need not start at offset 100 (hex).
;
start       proc    near
;
;Set the stack to 0020:0200, at the end of the unused interrupt vector
;area
;
   cli              ;Turn off interrupts so that we can't get
                    ;caught with the stack segment set the new way
                    ;and the stack pointer set the old way
   mov      ax,0020h
   mov      ss,ax
   mov      sp,0200h
   sti
;
;Move the rest of the boot routine to the unused
;vector area at 0020:0000
;
   mov      ax,moved_segment
   mov      es,ax   ;The extra segment will be the
                    ;destination segment
   mov      ax,cs
   mov      ds,ax   ;The code segment will be the source, since
                    ;we're relocating the program code
   call     self    ;Since we can't be sure where the routine is
                    ;loaded, we'll execute a call to the next
                    ;instruction to get the instruction pointer
                    ;pushed on the top of the stack
self:
   pop      si      ;Get the instruction pointer back off the stack
   add      si,((offset c1)-(offset self))  ;Adjust SI,
                    ;which will be the source for the move, to
                    ;point the the start of the working part of the
                    ;boot routine
   sub      di,di   ;The destination for the move will be 0020:0000
   mov      cx,256  ;Number of bytes to move
   rep movsb        ;Move the rest of the routine to 0020:0000
   jmp      moved_routine   ;Execute a far jump to the start of the
                    ; moved portion of the boot routine
;
;By the time the program reaches this point, the following code will
;have  been moved to 0020:0000.
;
c1:
   sub      ax,ax   ;0 is the number of the diskette reset function
   int      13h     ;Call BIOS to reset the disk
   mov      ax,program_segment
   mov      es,ax
   mov      ds,ax   ;Point the data and extra segments to point to
                    ; the segment in which the program will go
   mov      bx,offset start_point   ;point to the location at which
                    ; the program is to be loaded.  COM programs
                    ; must start at CS:100 (hex)
   mov      ch,1    ;Initial track #
lp1:
;
;Get the current track (4K bytes).  For 9 sectors/track, you will need
;to set a new disk parameter block-see Appendix A of the Tech Ref
;manual.Try to read the track-retry 10 times if the read attempt
;is not successful, and if still unsuccessful, enter an infinite loop.
;
   mov      bp,10   ;# times to retry
disk_retry_loop:
   push     bp
   sub      dx,dx   ;Head 0, drive 0
   mov      c1,1    ;Start at sector 1
   mov      ah,2    ;BIOS read function #
   mov      al,8    ;We want to read 8 sectors
   int      13h     ;Invoke BIOS to read this track
   pop      bp
;
;Check for error
;
   jnc      track_is_in
;
;If there is an error, reset disk and retry
;
   push     bp
   sub      ah,ah
   int      13h     ;reset disk
   pop      bp
   dec      bp
   jnz      disk_retry_loop
;
;If we've already retried many times, enter an infinite loop
;
error_loop:
   jmp      error_loop
;
;We've got this track
;
track_is_in:
   add      bx,1000h        ;Point to next load point
                            ;(4K farther on)
   inc      ch              ;Point to the next track
   cmp      ch,highest_track        ;See if we've read all the
   jbe      lp1             ;tracks in. If not, read the next track
;
;Set the stack for the application program.  Stack goes at the far end
;of  the program segment.  Turn off interrupts so we can't get caught
;between  setting SS and SP.
;
   cli
   mov      ax,program_segment
   mov      ss,ax
   mov      sp,0FC00h       ;Set this value to 0000h if you have
                            ;more than 64K of memory, to leave as
                            ;much room as possible free for the
                            ;program.  A lower value is required
                            ;if you have less than 64K so that the
                            ; stack will fit in memory
   sti
;
;The program is all loaded in.  Jump to the start of it.
;
   jmp      start_point
start       endp
cseg        ends
   end      start   ;Program is to begin at label START
```

LISTING 2: SAMPLE.ASM

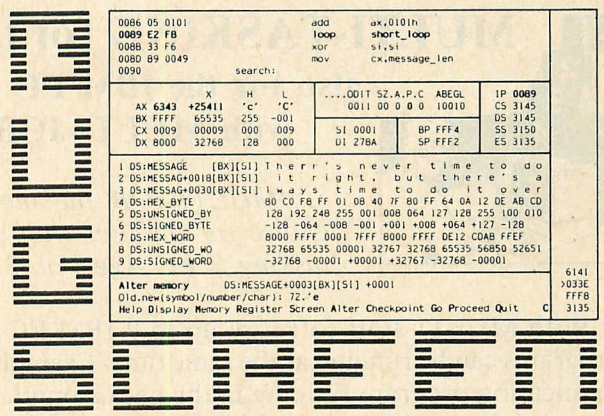
```
;
;Sample application program to be used with customized boot routine.
;
cseg        segment
   assume  cs:cseg,ds:cseg
   org     100h    ;COM files start at offset 100 (hex)
start       proc    far
;
;Output greeting string.
;
   mov      si,offset greeting_string
   call     string_out      ;Display initial message
;
;Program is finished-enter an infinite loop.
;
end_loop:
   jmp      end_loop
;
;Label strings.
;
greeting_string     db      0dh,0ah,'Congratulations!',0dh,0ah
   db       0dh,0ah,'You have booted a sample application program.'
   db       0dh,0ah,0
start       endp
```

```
;
;Subroutine outputs the string starting at SI.  String is terminated
;  with a 0 byte.
;
string_out proc    near
   lodsb           ;Get the next character
   and     al,al   ;Is it a 0 byte to mark end?
   jnz     not_end
   ret             ;If so, we're done
not_end:
   push    si
   sub     bh,bh   ;Display page 0
   mov     ah,14   ;Teletype-like character output function #
   int     10h     ;Display the character
   pop     si
   jmp     string_out
string_out endp
cseg       ends
   end     start   ;Program is to begin at label START
```

LISTING 3: MAKEBOOT.ASM

```
;
;Program to make a bootable disk, set to load and run the application
;  program upon booting.
;
stack   segment para stack 'STACK'
        db      64 dup('stack   ')
stack   ends
;
;Note: for convenience, both the boot routine and the application
; program  are contained in a single segment.  This limits the maximum
; size of the  application program to slightly less than 64K.
;
dseg    segment
boot_len        equ     512     ;Maximum length of boot routine
msg1    db      0dh,0ah,'File containing boot routine: $'
msg2    db      0dh,0ah,'Cant open boot routine file.',0dh,0ah,'$'
bootname        db      15,?,15 dup(?)  ;Temporary storage for boot
                                        ; routine file name
bootfcb db      37 dup(?)       ;File control block for boot routine
                                ;file
bootbuf db      boot_len dup(?) ;Buffer to hold boot routine
prog_len        equ     4096    ;Maximum length of application program
                                ;buffer (must be a multiple of 128)
msg3    db      0dh,0ah,'File containing application program: $'
msg4    db      0dh,0ah,'Application program file cannot be opened.'
        db      0dh,0ah,'$'
progname        db      15,?,15 dup(?)  ;Temporary storage for boot
                                        ;routine file name
progfcb db      37 dup(?)       ;File control block for boot
                                ;routine file
;
;Buffer for the application program.  The DUP factor should be large
;  enough to hold the whole program, but less than 64K.
;
progbuf db      prog_len dup(?)
progend label   byte    ;End of the program buffer
;
;Each 4-byte group describes the parameters to be used when formatting
;  a sector.  This format block is used for an 8-sector track.
;
format_block    db      0,0,1,2
        db      0,0,2,2
        db      0,0,3,2
        db      0,0,4,2
        db      0,0,5,2
        db      0,0,6,2
        db      0,0,7,2
        db      0,0,8,2
msg5    db      0dh,0ah,0dh,0ah,'Insert a blank disk in drive A:.'
        db      0dh,0ah,'This disk will be wiped out, so be careful!'
        db      0dh,0ah,'Strike any key to begin...',7,'$'
msg6    db      0dh,0ah,0dh,0ah,'The disk in A: is ready to boot.'
        db      0dh,0ah,'$'
dseg    ends
;
cseg    segment
        assume  cs:cseg,ds:dseg,es:dseg
start   proc    far
        push    ds
        sub     ax,ax
        push    ax      ;Set stack for return to DOS
;
;Set working data segment.
;
        mov     ax,dseg
        mov     ds,ax
        mov     es,ax   ;Set ES for the parse filename function
;
;Get the boot routine file control block set up.
;
        mov     dx,offset msg1
        mov     ah,9
        int     21h     ;Prompt for boot file name
        mov     dx,offset bootname
        mov     ah,0ah
        int     21h     ;Get boot file name
        mov     si,offset bootname+2    ;Skip the count bytes in
                                        ;bootname
        mov     di,offset bootfcb
        mov     al,1
        mov     ah,29h
        int     21h     ;Parse the filename into a file control block
;
;Open the boot routine file.
;
        mov     dx,offset bootfcb
        mov     ah,0fh
        int     21h     ;Open it
        and     al,al
        jz      read_boot
;
;Error in trying to open the boot routine file.
;
        mov     dx,offset msg2
        mov     ah,9
        int     21h     ;Notify of error
        jmp     done
;
;Read in the boot record.
;The record point field is set to the top of the file (record 0)
;by the open.
;
read_boot:
;
;First set load address.
;
        mov     dx,offset bootbuf
        mov     ah,1ah
        int     21h
;
;Now read it.
;
        mov     dx,offset bootfcb
        mov     cx,boot_len/128 ;# of 128-byte records to be read
        mov     ah,27h
        int     21h     ;Read it as a single block
;
;Now read in the application program.
;Get the application routine file control block set up.
;
        mov     dx,offset msg3
        mov     ah,9
        int     21h     ;Prompt for application file name
        mov     dx,offset progname
        mov     ah,0ah
        int     21h     ;Get application file name
        mov     si,offset progname+2    ;Skip the count bytes in
                                        ;progname
        mov     di,offset progfcb
        mov     al,1
        mov     ah,29h
        int     21h     ;Parse the filename into a file control block
;
;Open the application routine file.
;
        mov     dx,offset progfcb
        mov     ah,0fh
        int     21h     ;Open it
        and     al,al
        jz      read_appl
;
;Error in trying to open the application routine file.
;
        mov     dx,offset msg4
        mov     ah,9
        int     21h     ;Notify of error
        jmp     done
;
;Read in the application program.
;The record point field is set to the top of the file (record 0)
;by the open.
```

```
;
read_appl:
;
;First set load address.
;
        mov     dx,offset progbuf
        mov     ah,1ah
        int     21h
;
;Now read it.
;
        mov     dx,offset progfcb
        mov     cx,prog_len/128 ;# of 128-byte records to be read
        mov     ah,27h
        int     21h     ;Read it as a single block
;
;Now write on the disk in drive A:.
;
;Warn the user first!
;
        mov     dx,offset msg5
        mov     ah,9
        int     21h
;
;Wait for user to strike a new key to proceed.
;
        mov     ax,0c01h        ;DOS function empties key buffer before
        int     21h             ; reading next character
;
;Format and write the boot record via BIOS.
;
        sub     dx,dx   ;Side 0, drive 0
        sub     ch,ch   ;Track 0
        mov     cl,1    ;Sector 1
        mov     bx,offset format_block  ;Point to format parameters
        mov     ah,5    ;BIOS format track function
        mov     al,8    ;Format with 8 sectors
        int     13h     ;Format the track
;
;Write the track.
;
        sub     dx,dx   ;Side 0, drive 0
        sub     ch,ch   ;Track 0
        mov     cl,1    ;Start at sector 1
        mov     bx,offset bootbuf       ;Take input from here
        mov     ah,3    ;BIOS write sector function
        mov     al,1    ;Write 1 sector
        int     13h     ;Write the boot track
;
;Format and write each track of the application program in turn.
;
        mov     ch,1    ;Start at track 1
        mov     bx,offset progbuf       ;Point to start of application
                                        ; program
appl_loop:
        push    bx
        push    cx
        call    format_write_track
        pop     cx
        pop     bx
        inc     ch      ;Point to the next track
        add     bx,1000h        ;Point to the next section of program
                                ;code
        cmp     bx,offset progend       ;See if we're at the end of the
        jb      appl_loop               ; application program buffer
;
;Notify finished.
;
        mov     dx,offset msg6
        mov     ah,9
        int     21h
;
;Return to DOS.
;
done:
        ret
start   endp
;
;Subroutine to format and write the data starting at DS:BX to track CH.
;
format_write_track      proc    near
;
;First format the track.
;
        push    bx      ;Save the data to be written address
        sub     dx,dx   ;Side 0, drive 0
        mov     cl,1    ;Start at sector 1
        mov     bx,offset format_block  ;Point to format parameters
        mov     [bx],ch ;Set each of the format records to point to the
                        ; track to be formatted
        mov     [bx+4],ch
        mov     [bx+8],ch
        mov     [bx+12],ch
        mov     [bx+16],ch
        mov     [bx+20],ch
        mov     [bx+24],ch
        mov     [bx+28],ch
        mov     ah,5    ;BIOS format track function
        mov     al,8    ;Format with 8 sectors
        int     13h     ;Format the track
        pop     bx      ;Retrieve the address of the data to be written
;
;Write the track.
;
        sub     dx,dx   ;Side 0, drive 0
        mov     cl,1    ;Start at sector 1
        mov     ah,3    ;BIOS write sector function
        mov     al,8    ;Write 8 sectors
        int     13h     ;Write the track
;
        ret
format_write_track      endp
cseg    ends
        end     start   ;Program will begin at label START
```



C_to_dBASE

The MISSING LINK

C_to_dBASE is a new development tool that allows you to manipulate dBASE data and index files with programs written in the C language. C_to_dBASE provides more than 70 C language functions including:

- Functions to access and modify dBASE data and index files without using dBASE.
- Powerful C language functions for development of dBASE file management programs.
- A menu-driven sample application program that demonstrates the use of C_to_dBASE.
- Full source code in C.
- No royalties.

Whether you are a beginning or professional programmer, C_to_dBASE is a powerful tool for the development of data base applications. Only $150.00 (includes source code).

For More Information Or To Order Call:

800-922-0169

COMPUTER INNOVATIONS, INC.

980 Shrewsbury Avenue, Tinton Falls, NJ 07724

Prices are subject to change without notice.
dBASE is a trademark of Ashton-Tate.

CIRCLE NO. 181 ON READER SERVICE CARD

MULTI-TASKING for the IBM AT!

also for the IBM PC or XT with MULTI-JOB™

"MULTI-JOB the most cost effective choice for the user with a need for multi-tasking." PC Age Vol. 3.8

With MULTI-JOB software up to 9 IBM PC DOS compatible programs can be running at the same time. Example, have your communication program running in the background, and still be using your word processing, spreadsheet programs, etc., at the same time! The keyboard and screen can be assigned to any job with a simple keystroke. The remaining jobs will continue to run unattended. With the many different options, MULTI-JOB is a very powerful package.

* No special hardware is required.
* Allows priorities to be given between each job.
* Programs can be run simultaneously or one at a time.
* 30-day free trial period.

MULTI-JOB	$159.00
ELECTRONIC DISK	$ 49.00
SPOOL PROGRAM	$ 24.00
SET MEMORY UTILITY	$ 24.00

B&L COMPUTER CONSULTANTS, 7337 Northview, Suite B, Boise, ID 83704, (208) 377-8088.

MasterCard VISA

Dealer's inquiries are welcome. Call or write for a free catalog.

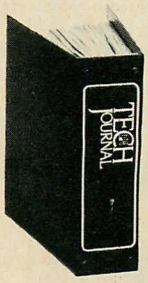
CIRCLE NO. 153 ON READER SERVICE CARD

SHELF CONSCIOUS?

Now you can organize your copies of

PC TECH JOURNAL

Now your magazines can be a handsome addition to your decor, well organized, and easy to find, thanks to these durable library-quality cases or binders. They're made of luxury-look leatherette over high-quality binder board. And both styles are custom-designed for this or any other magazine you save, with size, color and imprint selected by the publisher. FREE transfer foil included for marking dates and volumes.



Magazine binders
holds your issues on individual snap-in rods, combining them into one volume. $7.95 each; 3 for $22.50; 6 for $42.95. Mixed titles OK for quantity prices.

Open-back cases
store your issues for individual reference. $6.95 each; 3 for $19.75; 6 for $37.50. Mixed titles OK for quantity prices.

For faster service,
CALL TOLL-FREE
800-526-0790
(In NJ only 201-540-0445)

PC Tech Journal
P.O. Box 5120, Philadelphia, PA 19141

Please send: ☐ Cases ☐ Binders

TITLE	QUANTITY
PC Tech Journal	
Other:	

☐ PAYMENT ENCLOSED $_____.* Add $1.00 *per order* for postage and handling. Outside USA add $2.50 *per unit* ordered; send US funds only.

☐ CHARGE (Minimum $10):
☐ American Express ☐ MasterCard
☐ Visa

Card No._____ Exp. Date_____

Signature_____

Print Name_____

Address_____

City_____

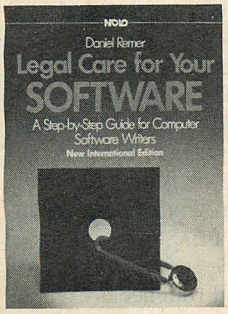
State/Zip_____

*Residents of PA add 6% sales tax.

A Good Legal Guide

. . . but it won't replace your lawyer

Legal Care for Your Software: A Step-by-Step Guide for Computer Software Writers *(New International Edition)*
Daniel Remer
(Nolo Press, Berkeley, CA, 1984)
172 pages, including about 85 pages of forms, $24.95



I am a lawyer. My business is advising people how to structure transactions that are related to computers. If there were a book that could replace my four years of expensive engineering education and my three years of law school at a tiny fraction of my hourly billing rate, I certainly would not tell you about it. I would buy each and every copy I could find and use all but one of them for mulch.

This book is not going to replace me. Even the author acknowledges that "this book is neither designed nor intended to replace lawyers." It is, however, well designed, well written and, perhaps most important, it fits my philosophy of how legal decisions should be made and reflected in documents. More about that in a moment.

First, a few warnings are in order. This book is not going to replace lawyers for three reasons: there is no mechanism for continuous updating to report current changes in the law; it cannot balance risks and negotiate based on those balances; it cannot cover all the nuances of every set of circumstances nor judge which nuance makes a difference.

The most significant aspect of this book is its "systems" design. A standard legal text treats one subject: one text for copyright, with perhaps a nod to trade secrets as a related topic; a second text for employment agreements, again with perhaps a mention of confidentiality agreements. *Legal Care for Your Software* is divided into chapters dealing with specific subjects (trade secrets, copyright, contracts, work for hire and assignments, evaluation agreements, license agreements, trademarks, patents, limiting liability, remedies, and international law), but is situation oriented: examples of real situations and how various legal techniques and principles bear on them are analyzed. As a result, it is much more likely that a layperson can appreciate the significance of the legal principles. The tradeoff is that if you are looking for guidance on a situation that is not included, it is difficult to ferret out the various issues from analogous situations.

The style is clear and direct, while probably a bit light for the taste of most law professors. The general pattern is to describe a situation, analyze the legal issues that are presented, and refer to the rather extensive collection of forms in the appendix. Forms can be very dangerous. Assume that every word is there for a reason and worry if you do not understand the reason.

For example, arbitration is recommended as a dispute resolution mechanism. The sample paragraph provides that the parties will settle their dispute "in accordance with the Expedited Rules of the American Arbitration Association then in effect." I would advise against using that language unless you have read a copy of those expedited rules (which are not in the book), and I would be cautious about the "then in effect" language, which allows the AAA to modify the contract without either party knowing what is being modified.

Legal Care for Your Software does suffer from some minor lapses. For example, I am accustomed to paying about $150 for a trademark search, so I was interested to read that there was a firm that would do such a search for $45 to $60. The author directed me to page 45. I finally found what I assumed to be the name of the firm in the footnotes on page 49 (which referred to a copyright search firm), and I gave up before finding where in the text footnote 5 appeared. I contacted the firm and their fee is actually $50 plus copying costs—but for a search of the federal records only. As the author points out, "The purpose of the search is to try to discover if anyone is already using the mark you have selected" and, as he also points out, there may be common law trademarks that have not been federally registered and obtaining a clean federal search is no guarantee of safety. The cost of the firm's "comprehensive search" is $125 plus copying costs.

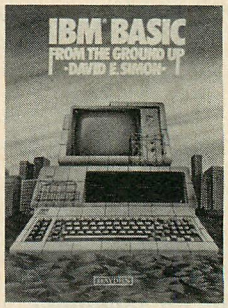
I do want to conclude on a positive note because this book really is a fine effort to make software law comprehensible and usable. I applaud the approach to adversary situations (such as negotiating a license or an employment contract) as a constructive process, with its goal being informed agreement. I think that is, in the long run, the way to get the best deal. The forms tend to be "fair," which probably makes them inappropriate as a first draft in negotiations that are anticipated to be fierce. They could serve in friendly situations or as guidelines, laying the groundwork for the final form.

In summary, I would not go into a high-stakes negotiation armed with nothing but this book. I would recommend it to sensitize the computer expert who is preparing a first encounter with the world of contracts. If I had a lawyer who could not tell his ASCII from his EBCDIC but seemed trainable, I would give him or her a copy as a present and consider the money well spent. And, if the publisher does not insist that I return the review copy, I will keep it on my bookshelf. For those who have never seen the condition of my bookshelf, take it on faith: that is high praise in and of itself.

—MAX STUL OPPENHEIMER

BOOK REVIEWS

IBM BASIC from the Ground Up
David E. Simon
(Hayden Book Company, Hasbrouck Heights, NJ, 1983)
306 pages, paper, $17.95



Covering a topic from the ground up is an ambitious undertaking in any field. The market is flooded with books that claim the ability to lead novices safely into the world of computer programming; this offering by David Simon can be counted among the successful.

No prior knowledge of computers is assumed of the intended reader. A reader armed with an IBM PC (minimum 64KB), single-disk drive, and monitor is equipped to tackle this book. Simon's goal is clear throughout: to help the reader learn, through the abundant use of both text and example, to program a computer in BASIC.

Much of the book is standard fare for introductory programming manuals. A starter's subset of BASIC is introduced gradually to include string manipulation, file usage, and rudimentary graphics (monochrome and color).

Several noteworthy features in this book stand out and distinguish it from the competition. First, it is aimed at and written for the absolute beginner. Simon is unusually conscientious and patient in guiding the reader through the fundamentals of the PC and its BASIC interpreter in a logical fashion.

Second, the author begins early in the book to promote good programming practices through the introduction of problem solving techniques and the use of pseudocode and structured programming. Simon has avoided the temptation to bombard the reader with the usual blizzard of computer buzzwords. Acronyms are used sparingly and are explained in language that is understandable to the beginner.

While pseudocode examples are typeset, the programs and program fragments included are dark, clear, and easy-to-read dot-matrix printouts. The programs are not presented as usual solutions to everyday problems, rather they are examples of particular language features and programming technique. Informal homework problems of varying degrees of difficulty abound and no answers are given.

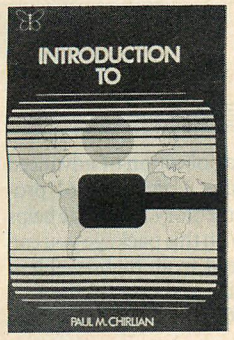
Another accomplishment worthy of particular note is Simon's clear explanation of debugging a program. Armed with the knowledge that many new users spend more time and generate more frustration searching for bugs than they do writing code, he explains basic debugging techniques in some detail. The judicious use of PRINT statements to find errors is discussed and an emphasis is placed on desk checking, which is the practice of working out the bugs in a piece of code before it is ever typed into the computer. The author has also included an example of the use of the built-in trace function (using TRON and TROFF).

The final particularly noteworthy feature of *IBM BASIC from the Ground Up* concerns standardization and portability. Like all BASIC interpreters in use today, IBM BASIC has many unique characteristics. In view of this fact, Simon clearly identifies, using boxed areas throughout the text, those language features not likely to be encountered in other brands of BASIC.

In summary, *IBM BASIC from the Ground Up* is a good introductory text for first-time users. It does not attempt to discuss IBM BASIC in exhaustive detail. It does what it sets out to do: cover the basics and cover them well.

—JEFFERY W. WILSON

Introduction to C
Paul M. Chirlian
(Matrix Publishers, Inc., Beaverton, OR, 1984)
187 pages, paper, $15.95



Sizing up an audience and estimating its needs is one of the most difficult tasks an author faces. Most *PC Tech Journal* readers use a personal computer and have at least a passing acquaintance with a computer language, usually BASIC. Many of these readers are exploring new languages, such as C and Pascal. Most of this learning is self-directed, requiring the gentle tutelage of a well-written text.

Introduction to C misses the mark for such readers because the author tries to reach too many people at once. His desired target audience consists of all users of the C language, regardless of their experience and the types of computers with which they are familiar.

In the preface, Chirlian states, "[This book] can be used by readers who wish to learn C even if they have minimal programming experience." He claims, "The various topics in C are presented in a simple manner in this book so as not to confuse the beginner." Finally, he says, "... all aspects of C are discussed so that this book could also serve as a reference for the experienced programmer." Well, not quite.

The book introduces the C language only by presenting its components, which is like teaching someone to drive by giving him an auto-parts catalog to read. The user is referred to compiler reference manuals (a topic for review in themselves) for details, such as how to compile the sample programs offered in the book. Examples from UNIX and one or two popular IBM PC implementations, such as those by Computer Innovations and Lattice/Microsoft would help the beginner enormously.

Structured programming, documentation, and debugging are covered

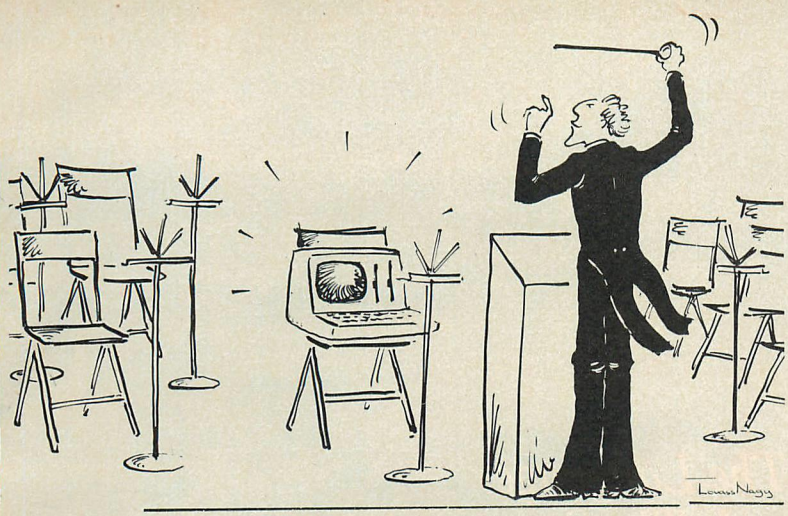
at a basic level. Homework problems appear at the end of each chapter; answers are not provided. A separate instructor's manual is mentioned in the preface. The book does not contain a glossary; tables of C keywords, C operators, and ASCII codes appear in the appendix. The index is adequate.

As a reference work, the book has two strikes against it. First, most users will choose for reference *The C Programming Language,* by Brian W. Kernighan and Dennis M. Ritchie (Prentice-Hall, 1978), which is the definitive work on the subject. Second, Chirlian's book lacks a complete and concise language description such as that found in Kernighan and Ritchie's work. When most experienced users go to a reference book, they look for tables, not prose, to answer their questions.

Presumably in an effort to keep production costs down, *Introduction to C* is largely devoid of white space and completely lacks explanatory diagrams. An astonishing number of typographical and spelling errors remain in the printed work, both in the text and in the sample programs. Aside from being distracting, these errors lead the reader to question the accuracy of the text.

With regard to accuracy, Chirlian is on safe ground in his description of C and the definition of common computer terms in all but one case: he chooses to contradict the conventional definition of the term *forward reference.* Well, at least he spelled it correctly.

—JEFFERY W. WILSON



Would you hire an entire band when all you need is one instrument? Of course not.

So why use a whole orchestra of computers when all you need is one to develop software for virtually any type of micro-processor?

The secret? Avocet's family of cross-assemblers. With Avocet cross-assemblers you can develop software for practically every kind of processor — *without having to switch to another development system along the way!*

Cross-Assemblers to Beat the Band!

Development Tools That Work

Avocet cross-assemblers are fast, reliable and user-proven in over 4 years of actual use. Ask NASA, IBM, Xerox or the hundreds of other organizations that use them. Every time you see a new micro-processor-based product, there's a good chance it was developed with Avocet cross-assemblers.

Avocet cross-assemblers are easy to use. They run on almost any personal computer and process assembly language for the most popular microprocessor families.

Your Computer Can Be A Complete Development System

Avocet has the tools you need to enter and assemble your soft-ware and finally cast it in EPROM:

VEDIT Text Editor makes source code entry a snap. Full-screen editing plus a TECO-like command mode for advanced tasks. Easy installation - INSTALL program supports over 40 terminals and personal computers. Customizable keyboard layout. CP/M-80, CP/M-86, MSDOS, PCDOS $150

EPROM Programmers let you program, verify, compare, read, display EPROMS but cost less because they communicate through your personal computer or terminal. No personality modules! On-board intelligence provides menu-based setup for 34 different EPROMS, EEPROMS and MPUs (40-pin devices require socket adaptors). Self-contained unit with internal power supply, RS-232 interface, Textool ZIF socket. Driver software (sold separately) gives you access to all programmer features through your computer, lets you download cross-assembler output files, copy EPROM to disk.

Model 7228 Advanced Programmer — Supports all PROM types listed. Super-fast "adaptive" programming algorithm programs 2764 in 1.1 minutes.

Model 7128 Standard Programmer — Lower-cost version of 7228. Supports all PROM types except "A" versions of 2764 and 27128. Standard programming algorithm programs 2764 in 6.8 minutes.

Avocet Cross-assembler		Target Microprocessor	CP/M-80	CP/M-86 IBM PC, MSDOS**
XASM04	*NEW*	6804	$ 250.00	$ 250.00
XASM05		6805	200.00	250.00
XASM09		6809	200.00	250.00
XASM18		1802/1805	200.00	250.00
XASM48		8048/8041	200.00	250.00
XASM51		8051	200.00	250.00
XASM65		6502/65C02	200.00	250.00
XASM68		6800/01, 6301	200.00	250.00
XASM75		NEC 7500	500.00	500.00
XASM85		8085	250.00	250.00
XASM400		COP400	300.00	300.00
XASMF8		F8/3870	300.00	300.00
XASMZ8		Z8	200.00	250.00
XASMZ80		Z80	250.00	250.00
XMAC682	*NEW*	68200	595.00	595.00
XMAC68K	*NEW*	68000/68010	595.00	595.00

Model 7956 and 7956-SA Gang Programmers — Similar features to 7228, but program as many as 8 EPROMS at once. 7956-SA stand-alone version copies from a master EPROM. 7956 lab version has all features of stand-alone plus RS-232 interface.

EPROM: 2758, 2716, 2732, 2732A, 2764, 2764A, 27128, 27128A, 27256, 2508, 2516, 2532, 2564, 68764, 68766, 5133, 5143. **CMOS:** 27C16, 27C32, 27C64, MC6716. **EEPROM:** 5213, X2816A, 48016, I2816A, 5213H. **MPU (w/adaptor):** 8748, 8748H, 8749, 8749H, 8741, 8742, 8751, 8755.

7228	**Advanced Programmer**	**$ 549**
7128	**Standard Programmer**	**429**
7956	**Laboratory Gang Programmer**	**1099**
7956-SA	**Stand-Alone Gang Programmer**	**879**
GDX	**Driver Software**	**95**
481	**8748 Family Socket Adaptor**	**98**
511	**8751 Socket Adaptor**	**174**
755	**8755 Socket Adaptor**	**135**
CABLE	**RS-232 Cable (specify gender)**	**30**

HEXTRAN Universal HEX File Converter — Convert assembler output to other formats for downloading to development systems and target boards. Also useful for examining object file, changing load addresses, extracting parts of files. Converts to and from Intel, Motorola, MOS, RCA, Fairchild, Tektronix, TI, Binary and HEX/ASCII Dump formats. For CP/M, CP/M-86, MSDOS, PCDOS $250

Ask about UNIX.

68000 CROSS-ASSEMBLER — With exhaustive field testing completed, our 68000 assembler is available for immediate shipment. XMAC68K supports Motorola standard assembly language for the 68000 and 68010. Macros, cross-reference, structured assembly statements, instruction optimization and more. Linker and librarian included. Comprehensive, well-written manual.

To find out more, call us toll-free.

1-800-448-8500

(in the U.S. Except Alaska and Hawaii)

VISA and Mastercard accepted. All popular disc formats now available --please specify. Prices do not include shipping and handling -- call for exact quotes. OEM INQUIRIES INVITED.

*Trademark of Digital Research **Trademark of Microsoft

AVOCET SYSTEMS INC.™

Sales and Development:
10 Summer Street
P.O. Box 490, Dept. 185-PCT
Rockport, Maine 04856
(207) 236-9055 Telex: 467210 AVOCET CI

Corporate Offices:
804 South State Street
Dover, Delaware 19901

Serious Software For The C Programmer From Lifeboat.™

Lattice® C Compiler: *The serious software developer's first choice.*

Selected for use by IBM,® Texas Instruments, Wang,® MicroPro,® Ashton-Tate,™ IUS/Sorcim,® Microsoft® and Lotus™ to name a few of the many. Why? Lattice C is clearly the finest 16 bit C compiler available today.

—Renowned for speed and code quality.
—Fully compatible with the C standards set forth by Kernighan and Ritchie.
—Four memory model options offer you unsurpassed control and versatility.
—Superior quality documentation.
—Now includes automatic sensing and use of the 8087 chip.
—Widest selection of supporting add-on packages.

Halo™: *A graphics development package rapidly emerging as the industry standard.*

—140 graphics commands including plot, line, arc, box circle and ellipse primitives, bar and pie charts; pattern fill and dithering commands.
—New: multiple viewports and "stroke text" for angling, scaling and filling text.

C Food Smorgasbord™: *This beautifully written collection of C functions is a valuable time saver.*

—Library includes a binary coded decimal arithmetic package, level 0 I/O functions, a terminal independence package, IBM PC ROM BIOS access functions and much more.

Pmate™: *The premier editor for the programming professional.*

Pmate is a full screen editor with its own powerful macro command language:

—Perform on screen row and column arithmetic, alphabetize lists, translate code from one language to another, call up other macros.
—Customize Pmate almost any way you like.
—Contains 10 auxiliary buffers for storage of macros, text, subroutines.
—An "undo" feature allows the programmer to retrieve whole series of deleted items.

Additional C Tools Available From Lifeboat:

Panel™: Screen formatter and data entry aid.
Lattice Windows™: Windowing utility; create "Virtual Screens."
Plink-86™: The popular linker; includes extensive overlay capabilities.
Pfix86™: Dynamic debugging utility.
Pfix86 Plus™: Symbolic debugger with capacity to debug overlays.
Btrieve™: Database record access/retrieval library.
Phact: Multikeyed ISAM C-Function library.
Fabs: Fast access B-tree database function library.
Autosort: Fast sort/merge utility.
ES/P: 'C' program entry with automatic syntax checking and formatting.
Greenleaf Functions™: Library of over 200 popular C functions.
And much more.

YES! Please rush me the latest FREE Lifeboat™ catalog of C products.

*Company Name*_____ *Business Phone*_____

*Name*_____ *Title*_____

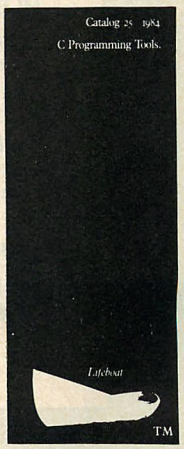
*Address*_____

*City*_____ *State*_____ *Zip*_____

Please check the category where Lifeboat can best help you:

☐ *Software development* ☐ *Corporate* ☐ *Education*
☐ *Dealer/distributor* ☐ *Government* ☐ *Other*_____

Call Direct: 1-800-847-7078 (In NY State: 212-860-0300)
Return coupon to: Lifeboat Associates™
1651 Third Avenue, New York, NY 10128. *TJ112*



CIRCLE NO. 139 ON READER SERVICE CARD

The C Compiler Thousands Rely On

C-86™

NEW-IMPROVED Version 2.2 Compiles 25% Faster
IBM-PC AT Support

When the going gets tough, Optimizing C86 comes through time and time again. C86 is a highly dependable C compiler that has been optimized through the years to provide the best combination of reliability, speed, and performance.

FAST, IN-LINE 8087/80287 SUPPORT

Now you can take full advantage of 8087/80287 capabilities, allowing your programs to run many times faster than possible with other C compilers. Plus the source code to all routines is included, so you have complete control over all functions.

MORE OF THE FEATURES YOU WANT

- **SOURCE** is provided to all libraries for total programming control. The source includes a set of standard UNIX routines plus many DOS specific functions.
- **SPECIAL IBM-PC LIBRARY** including communication, screen, and keyboard handling functions.
- **COMPATIBLE WITH WIDELY AVAILABLE LIBRARIES** such as HALO screen graphics and many, many others (call for list).
- **TOPVIEW SUPPORT LIBRARY** provides windowing capabilities.
- **SPEED OPTIMIZATION** — there's always room to tighten your code, and Computer Innovations has the tools to help. For example, *PROFILER-86* helps identify key areas for optimization.

TECHNICAL SUPPORT, NOBODY DOES IT BETTER

Computer Innovations has earned a reputation for providing customer support that is **unequalled** in the industry. This includes a user's group, an on-line bulletin board, and a user's newsletter.

JOIN THE THOUSANDS OF PROGRAMMERS WHO TRUST AND RELY ON C86

For Further Information Call 800-922-0169.

Technical Assistance Call (201) 542-5920.

Computer Innovations features a full line of C products including **C-to-Dbase** (Dbase development tool) and **Introducing C** (C Interpreter Language Learning System). Call or write for a product profile.

COMPUTER INNOVATIONS, INC.

980 Shrewsbury Avenue, Tinton Falls, NJ 07724

For Further Information Call

800-922-0169

Technical Assistance Call (201) 542-5920

Shrink-wrapped Magazines

Bizarre as it sounds, the tear-me-open license agreement used by software publishers may be migrating to magazine publishers.

Software publishers take great time and expense to shrink-wrap a program in order to create the appearance that it is being licensed rather than sold. Book and magazine publishers have never gone to such trouble, but a new issue is posed by magazines devoted to listings of source codes.

Try to imagine the racks of your local bookstore with its copies of *Penthouse* and *Nibble* wrapped in plastic, with the latter being easily identified by its red warning label on the outside that says "DO NOT OPEN THIS PACKAGE UNLESS YOU ARE WILLING TO SUBMIT TO THE FOLLOWING TERMS. . . ." The image does not come easily. Yet publishers of magazines that provide source code listings face the same problem with regard to copyright as do publishers of software.

In the case of software, the purpose of the so-called *tear-me-open license agreement* is to allow the publisher to argue to a court that the purchaser read the fine print and made a conscious decision to accept its terms before unwrapping the package. This extraordinary fiction is the result of a compromise between the computer publisher's lawyers (who would probably prefer to obtain individually signed agreements before the disk is delivered) and its marketing director (who would probably prefer that the lawyer stop mucking up the packaging). Why does the lawyer prefer that the transaction be a license rather than a sale?

A provision of the Copyright law (17 USC Section 109) gives the owner of a lawful copy of a copyrighted work the right to "sell or otherwise dispose of the possession of that copy." The purchaser of a book can read it and then lend it to a friend without violating the author's copyright; the owner of a copyrighted painting can hang it on his wall for friends to see, without violating the artist's copyright.

It is feared that the purchaser of computer software could load the program into several machines (or even worse, into one CPU serving multiple users). Some software publishers permit such activities, but charge a multiuser fee. It is not, however, always possible to determine who the ultimate user will be. Once the package has been sold, Section 109 permits the new owner to resell it without the permission of the copyright owner.

It is an open issue whether the tear-me-open license is in fact a license and, if so, whether it is enforceable as written. The fact that one party calls a transaction a license does not make it one, particularly when it takes place in a context that resembles a sale.

The state of Louisiana tried to help the software publishing industry by passing a statute saying that when a publisher calls a transaction a license, that's what it is. That statute applies only to transactions in intrastate commerce (that is, those that take place entirely within Louisiana, which might mean that not only the transfer but also the creation of the software had to have taken place in Louisiana). To the extent the application of the statute would defeat the provisions of the federal copyright law, it is questionable if the statute would be enforceable at all.

Computer lawyers are, generally, not entirely happy about the tear-me-open license, and, whenever two or more meet, they ask each other if there is a better solution. To date, it appears that the shrink-wrap package and tear-me-open license is the best solution to the fear of unrestricted use of a single copy of a program. Whether that fear is sufficiently justified to warrant the effort is another matter.

Section 117 of the Copyright Act gives the owner of a copy of a computer program the right "to make or authorize the making of another copy or an adaptation of that computer program provided that such a new copy is created as an essential step in the utilization of the computer program in conjunction with the machine and in no other manner. . . ."

When Congress revised the Copyright Act in 1980, it had learned from the CONTU (Commission on New Technological Uses) that computer programs, unlike books, technically had to be copied into the computer's memory in order to be used. Congress was concerned that someone might take the position that the purchaser of a computer program disk could not run it without infringing the author's copyright.

Unfortunately, the words Congress chose can be read to permit more. For example, could the statute not be read to permit the purchaser of a magazine to type the source code listings into his or her machine, then save the program on a disk, all as "essential steps" in his utilization of the program? Should this be permitted? The owner of a book could not type the book into his computer and store it on disk without violating the author's copyright.

A recent case from the federal district court (the lowest court in the federal system) exemplifies some of the issues that may arise. In Micro-Sparc, Inc. *v.* Amtype Corporation, 592 F. Supp 33 (U.S.D.C., Mass. 1984), the defendant (Amtype) offered what it termed a "typing service" to purchasers of *Nibble* magazine. (Recall that what one party calls a transaction is not dispositive.)

The service Amtype performed was to type the source code listings from the magazine into its computer (which, assuming it had bought a copy of the magazine, it clearly had the right to do under Section 117), store the programs on a disk (something it arguably had the right to do under Section 117), then copy and sell the disk to people who signed a declaration that they had bought the magazine.

Micro-Sparc, the owner of the copyright on the programs (which also sold the programs on disk at a higher price), sued, alleging infringement of its copyright on the February and March

1984 issues. Interestingly, it did not raise the fundamental issue of whether the defendant, authorized or not, had the right to store the program on disk. While it is an "essential step in the utilization of the computer program" somehow to input the printed program to the machine's memory, that had already been accomplished by typing—arguably, saving the program to disk was not "essential" and therefore not protected by Section 117. Micro-Sparc may have chosen not to press this point because it is obviously intended that the magazine purchasers would store the programs on a diskette, and Micro-Sparc did not want to confuse anyone about that implicit authorization.

The defendant's position was virtually unassailable:

- An owner of a copy of the magazine could, under Section 117, not only make a copy of the programs, but could also "authorize" the making of a copy. The plaintiff conceded that the purchasers of its magazine owned a copy of the programs appearing in the magazine (and thus were entitled to the benefits of Section 117).
- Since, as the defendant asserted and the plaintiff did not contest, the "typing service" was provided only to owners of the magazine, and since the plaintiff had "authorized" the defendant to make the copy, no infringement took place.
- The fact that the defendant was copying and selling plaintiff's copyrighted works in competition with the plaintiff was irrelevant.

The district court held that the defendant had infringed the plaintiff's copyright, construing the purpose of Section 117 as to allow the input of a program into a computer's memory. In the court's view, the purchaser of the magazine could input the program without violating the copyright but, since the defendant did not input the program, its copying was not protected by Section 117. The court did not explain why the word "authorize" does not cover the defendant's activity.

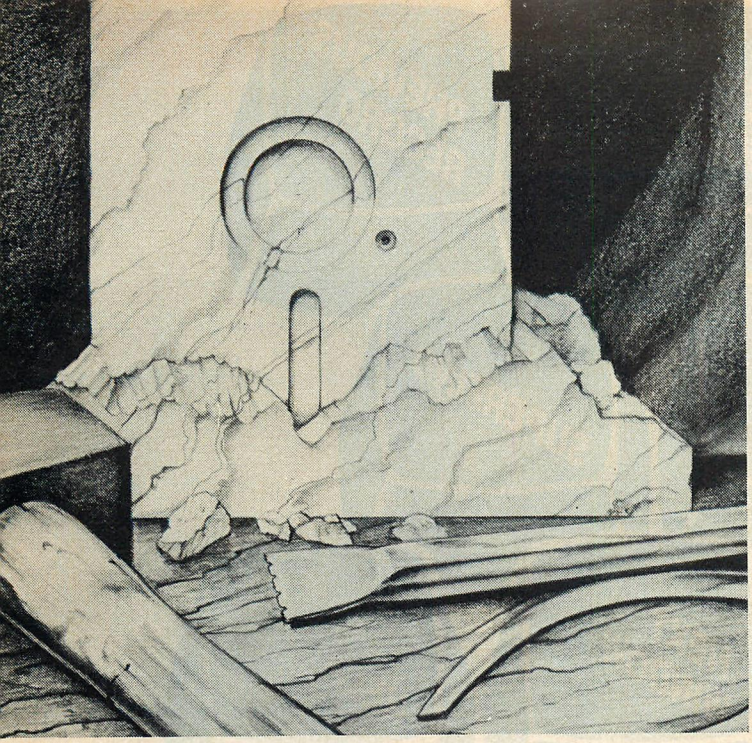
Beginning with the July 1984 issue, Micro-Sparc adopted the approach used by disk publishers. The fine print on the table of contents page of the magazine now says "Micro-Sparc Inc. is the owner of all rights in the computer programs printed in this magazine. To allow for their use by the purchaser of this magazine, Micro-Sparc Inc. grants to such purchaser, only, the Limited License (1) to enter these programs into the purchaser's computer, and (2) to place such computer programs on a diskette for personal use. Any other use . . . is expressly prohibited."

Whether this fine print inside the magazine can serve the same purpose as fine print shrink-wrapped on the outside of a disk package remains to be decided; there is a critical distinction between the two.

The argument fundamental to the success of the tear-me-open license (if it succeeds) is that the purchaser knew (and therefore agreed to) the terms before he made the agreement. Once an agreement is made, one party cannot unilaterally change its terms. It is quite a different argument that a customer can buy a magazine, pay the cover price, and only later be told that he must agree to further terms.

The result may be shrink-wrapped magazines. Let us hope that it will be limited to computer magazines.

Max Stul Oppenheimer, PC, is a partner in the law firm of Venable, Baetjer & Howard located in Baltimore.



"In the art of programming the difference between greatness and mediocrity is often the quality of the artist's tools."

POLYOVERLAY

Make Maximum Use of Memory with Overlay Code. Creates optimal memory overlay structures, in terms of utilizing a minimum of total physical memory, while it builds a batch file to automatically drive an overlay linker, for creating efficiently overlayed executable code. Creates optimum overlay structures for any modular language including C, FORTRAN, Pascal, Ada and even BASIC. Essential for porting mainframe code too large to fit into personal computer's memory all at one time. Requires MS-DOS 2.0, and Microsoft's Linker Version 2.4 Shipped with Users Manual **$99**

Praise From Professionals

"PolyLibrarian is a powerful tool for serious hobbyists and professional programmers. It is a thing of beauty; a work of art. The documentation is excellent, the large selection of commands are intuitive to learn and easy to use, and the program itself provides a service of inestimable value. It is refreshing to see a tool of this caliber available for serious programmers. The design of this program shows foresight and ingenuity. It sets the human engineering standard for programming utilities. If all of a programmer's tools were so simple to work with, better programs could be developed in less time and with fewer headaches. *Bravo, Polytron!* I look forward to your next product."

Dan Rollins,
Programmer and Author writing in PC Age

"PolyLibrarian is an extensive, friendly utility . . . an excellent tool for serious programmers. It combines professional quality with a flexible user interface."

Greg Estes
Editor, Programmers Journal

"I am thoroughly pleased [PolyLibrarian] is definitely one of the best products of its type that I have ever used on any system."

An Unsolicited Comment from
Steve Kauffman
Consulting Engineer

POLYTRON C LIBRARY 1

A Library of High-Performance Functions for Lattice C Compiler Users. Over 50 significant routines more useful than common library functions. The library includes UNIX-like utility functions, file management functions, more efficient and flexible I/O routines, and executive functions. This is a powerful, professional-quality C library that is easy to use and designed to increase your productivity. Assembly and C source code included. Requires Lattice C Compiler Includes Users Manual **$99**

POLYLIBRARIAN

The Object Module Library Manager for MS-DOS™. PolyLibrarian lets you create a single library from related Objects Modules. The IBM PC Linker will then select only the modules necessary to produce an executable file. PolyLibrarian can: Create, organize, reconstruct, dissect, and compress libraries of object code modules. Add, delete and replace an object module. Extract a module from a library to create an OBJ file. Change public and external names. Modules are time-stamped within the library. Supports full path names. Supports Microsoft's original and latest Linker (Version 2.4). Includes installation program to customize options to your specific needs. Features three Modes of Operation (Interactive, Command File, Command Line) that can be freely intermixed. Surpasses performance of other primitive librarians. Features a highly functional user interface that uses simple meaningful key words. Requires 128K RAM, MS-DOS (PC-DOS) 1.1 or 2.0. Compatible with any compiler or assembler that uses the MS-DOS (PC-DOS) Linker. Shipped with comprehensive Users Manual including step-by-step tutorial for **$99**

POLYLIBRARIAN II

Extends PolyLibrarian's power to Intel™ and Digital Research™ Format Libraries. PolyLibrarian II is an enhanced version of PolyLibrarian. In addition to all the features and functions listed above for PolyLibrarian, this version adds a new dimension of productivity for programmers using Intel and Digital Research Language Products. Compatible with any Intel Format Library and any Microsoft Format Library. Compatible with Intel's Link 86 under MS-DOS. Compatible with Digital Research Linker under MS-DOS **$149**

POLYMAKE

The Intelligent Program Builder & Maintenance Tool for MS-DOS. PolyMake frees you from the need to remember which files depend on others and which files have been modified. PolyMake will: Remember the exact sequence of operations necessary to make a new version of a program. This will significantly reduce developement time, prevent bugs & manage large software projects. Automatically invoke your compiler, assembler, linker, librarian, or do whatever is necessary to bring all dependent files up to date. Compare the date and time of all relevant files and use internal rules to rebuild a program or complete software system. Rules can be modified and expanded. Extensive debug capabilities help construct efficient make files. Includes features superior to UNIX Make. Fully automatic operation under MS-DOS 2.0. Full path name capabilities under MS-DOS 2.0. Requires 128K. Shipped with a comprehensive Users Manual **$99**

POLYFORTRAN TOOLS 1

Four Powerful Utilities in One Package. For developing or enhancing software or porting mainframe software to personal computers these easy-to-use tools will become indepensible for the professional FORTRAN programmer. Includes: 1. **FORTRAN-XREF** for developing comprehensive cross-reference maps of FORTRAN source code. Mainframe users expect this kind of utility. Now personal computer users have X-REF power. 2. **FORTRAN-EXTRACT** extracts the MAIN routine as well as all SUBROUTINES, FUNCTIONS and BLOCKDATAS from FORTRAN source code. It is unsurpassed for breaking-up large codes into manageable modules that can be individually modified, compiled, added to libraries and finally linked into a single executable code. 3.**FORTRAN-CLEANUP** performs three critical functions to improve FORTRAN source code readability: A. Renumbers FORTRAN source code statements within each routine to begin at a user specified starting number; B. Adds spaces around delimiters and key words; C. Performs indentation for DO loops, IF...Then and similar structures. If you are becoming weary of cleaning up "sloppy" code then you need FORTRAN-CLEANUP. 4. **FORTRAN-SCREEN** lets you show off your software. This is a complete set of FORTRAN callable screen attribute and cursor position control functions for creating user friendly, interactive menus and displays. Requires MS-DOS 2.0, 256K, One Disk Drive, Compatible w/hard disk. Shipped with Users Manual **$179**

To Order or Request Literature Call

1-800-547-4000

Ask For Dept. 310
VISA & MasterCard Accepted
Or Send Checks, P.O.'s To:
Polytron Corporation DS-310-F1
P.O. Box 787, Hillsboro, OR 97123
Add $1.25 Shipping Charge For Each Product Ordered.

CIRCLE NO. 268 ON READER SERVICE CARD

TM

POLYTRON

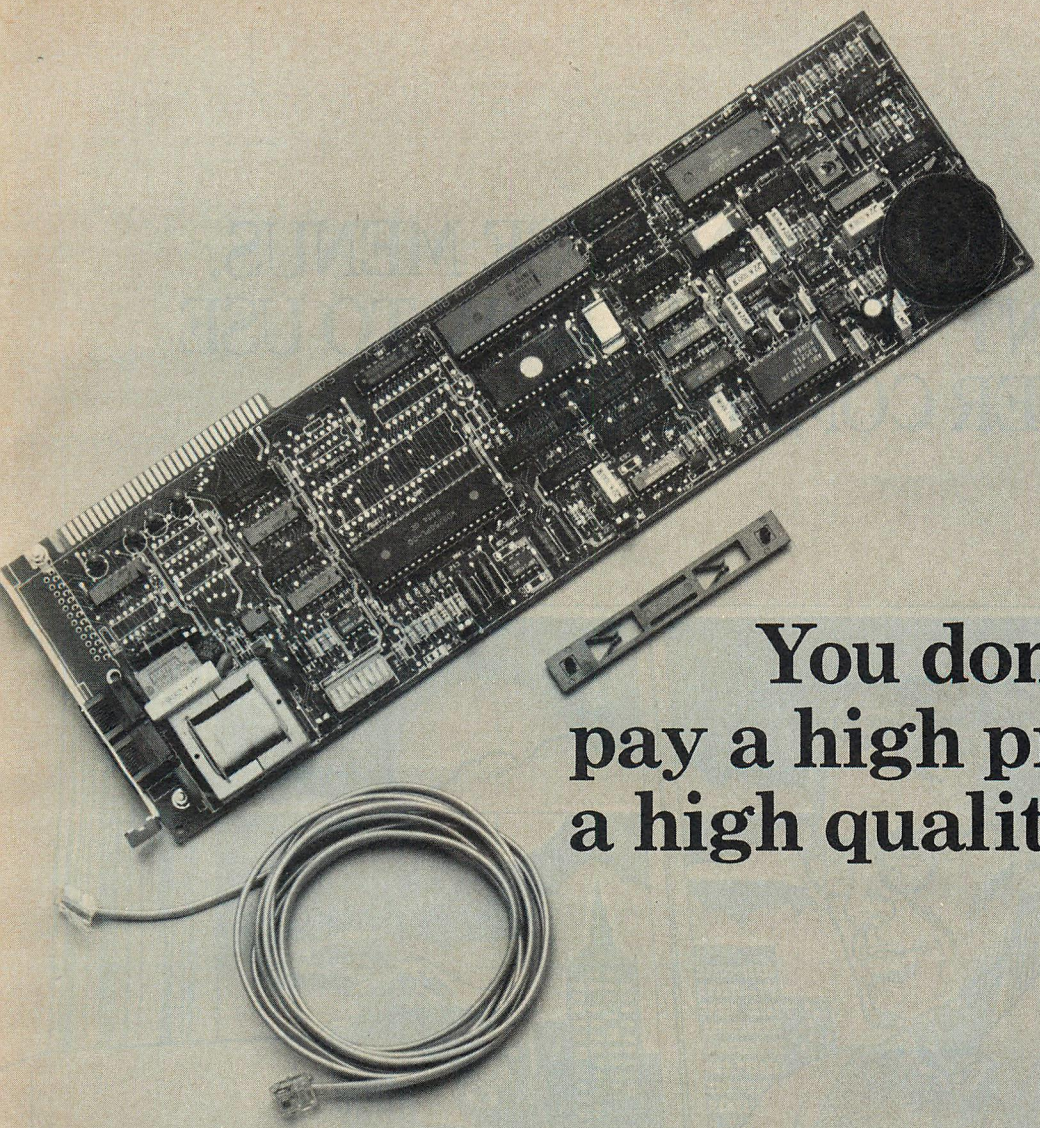
Fifteen user defined Flex-Keys provide single keystroke data entry and database commands, making DataFlex one of the most user-friendly database systems available. DataFlex's powerful program generator processes easy to create screen "images" to quickly define applications and produce ready to run, menu-driven software that's perfectly tailored to your needs. From simple single file applications to complex multi-file/ multi-key/multi-indexed systems, nothing does it faster, better or easier than DataFlex. Even multiple page HELP screens are produced with a one word command and called automatically at literally, the touch of a button. Call or write today for our latest literature and a list of existing DataFlex applications.

DATAFLEX™

DATA ACCESS CORPORATION.
8525 SW 129 Terrace, Miami, FL 33156 (305) 238-0012
Telex 469021 DATA ACCESS CI

compatible with MSDOS, PC-DOS, CP/M, CP/M-86, MP/M-86, TurboDOS, Novell Sharenet, PC-Net, Molecular N-Star, Televideo MmmOST, Action DPC/OS, Onminet, IBM PC w/Corvus and OSM Muse.
MSDOS is a trademark of Microsoft. CP/M and MP/M are trademarks of Digital Research. DataFlex and FlexKeys are trademarks of Data Access Corp.

CIRCLE NO. 146 ON READER SERVICE CARD



IBM PC USERS:

You don't have to pay a high price to get a high quality modem.

Our QIC-03 300/1200 internal modem is better than Hayes Smart Modem. Here is the comparision:

	QIC-03 300/1200	Hayes 1200B
Fully compatible with Hayes Command	Yes	Yes
300/1200 baud auto dial, Auto answer	Yes	Yes
Plugs inside IBM PC and compatibles	Yes	Yes
Large-Scale-Integration "Modem on a chip" for high reliability	Yes	No
Low heat dissipation	Yes	No
Low power consumption	Yes	No
On board speaker	Yes	Yes
Software volume control for speaker	Yes	No
Two plug in jacks for voice/data	Yes	Yes
One year warranty	Yes	Yes
Price	**$275**	**$595**

Compare Software Too!

	QIC COM	HAYES SmartCom
Easy to use "fill-in-the-blank" set up	Yes	No
Auto dial, Auto log-on	Yes	Yes
Supports 132 column display	Yes	No
Text and program file transfer (XMODEM)	Yes	Yes
Interrupt driven for high speed communication	Yes	No
Runs any DOS program without disconnecting	Yes	No
Powerful script file and action file set up for custom tailor applications	Yes	No
XON/XOFF Protocal for flow control	Yes	Yes
Terminal emulation	DEC VT100, IBM 3101 or others	No
Price	**(Free with purchase of modem)**	**$45.00**	**$175.00**

One Year Guarantee
No Risk. All our products are guaranteed for one year. And remember, if for any reason, you are not completely satisfied within 30 days, you can return it for a full refund. Check out our competition and find out which is best. We know better.

QIC RESEARCH INCORPORATED
528 Valley Way
Milpitas, CA 95035
408/942-8086

REAL MEN DON'T USE MENUS. I WANT TO KNOW HOW TO USE POWER COMMANDS.

HERE'S HOW:

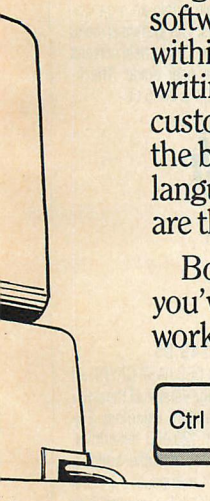
FRAMEWORK SOFTWARE

Framework™ has elegant menus that are handy for beginners and occasional users, but are easily bypassed once you know your way around the program.

We're going to show you the simplicity and speed of using power commands to create and use our unique automatic outline.

This fancy keywork is typical of the power commands for all of the powerful integrated functions of Framework: word processing, spreadsheet, graphics, data management, and telecommunications, as well as running other sophisticated software such as dBASE III™ within Framework. And for writing macros or creating custom programs with FRED,™ the built-in programming language, power commands are the only way to go.

Boot Framework, and you've created the Framework desktop.



Ctrl C 0 **Press the Ctrl and C keys together, then press the 0 key.** An outline appears on the screen.

↵ **Type Outline (or any title you like) and press the Return key.** The outline is titled.

\+ **Press the grey + key.** You move to heading 1.0.

↵ **Type Main Point 1 then press the Return key.** Heading 1.0 is titled.

↓ **Press the (down) arrow key.** You've moved to subhead 1.1.

Ctrl C E **Press the Ctrl and C keys together, then the E key.** A new heading is inserted under 1.1.

Ctrl + **Press Ctrl and the grey + key.** A subhead is added to the heading you just created. (Under any of these heads or subheads, you can be writing text, creating spreadsheets, generating graphics, etc.)

Ctrl C S **Press the Ctrl and C keys together, then the S key.** A spreadsheet frame is created as the second subhead.

F9 **Press the F9 key.** There's the spreadsheet you created, full screen.

F9 **Press F9 again.** Back to the outline.

Ctrl D C **Press the Ctrl and D keys together, then the C key.** Your desktop is cleaned up.

Now, how's that for a power trip. And you did it by following a few simple directions. Amazing. In just seconds you were in control of a powerful creative tool. Which is the way we think software should work.

For a dealer near you call (800) 437-4329, ext. 2230. In Colorado, (303) 799-4900, ext. 2230.

Software from

ASHTON-TATE™

We'll put you in control.

TECH BOOK

A Special Section for Product and Service Listings

Accessories/Supplies

IBM STYLE BINDERS, SLIPS

Complete program packaging line. D-ring cloth binders, slip cases, floppy pages, game portfolios. Continuous paper with three large holes, 20 lb. to go in binders. Blank disk envelopes. Function key cards tell user your F1-F10 meanings. Call, write for prices. Catalog. Fast service, low prices

ANTHROPOMORPHIC SYSTEMS LIMITED
376 E. St. Charles Road
Lomobard, IL 60148
(312) 629-5160

RESET BUTTON FOR IBM PC

The button IBM forgot. A good reset button is a basic tool for PC (PC/XT) programmers. When your machine is hung up, a touch of the button will quickly reboot your computer. This kit installs easily and mounts out of the way behind your machine. Easy instructions included. Send $19.80 (add 6% in MN).

RESET CORPORATION
P.O. Box 14809
Minneapolis, Minnesota 55414
(612) 489-5892

Hardware/Add-On Boards

FIXED DISK BIOS/BOOT

fiXT boots from DATAMAC, DAVONG, XEBEC, PERCOM, GREAT LAKES, ZOBEX, others. Adds XT-like BIOS interface for your disk to IBM PC or COMPAQ. Plug-in installation. DOS 2.0/3.2/reqd. Specify controller model with order $70 + $3 shpg. + tax. MC/VISA (optional volume support at additional cost.)

GOLDEN BOW SYSTEMS
P.O. Box 3039
San Diego, CA 92103
(619) 298-9349

DT2801 SERIES ANALOG I/O

Plug-in data acquisition boards with 8DI/16SE analog inputs, high or low level programmable gains, 2 analog outputs, 16 lines of digital I/O, DMA, on-board clock, and on-board microprocessor. Optional software subroutines and screw terminal panels.

DATA TRANSLATION INC.
100 Locke Drive
Marlboro, MA 01752
(617) 481-3700

TOTAL PC&XT INPUT/OUTPUT

ESE introduces two extremely powerful communications adapters for the PC&XT & compatibles. One is a unique multi-interrupt RS-232 serial add-on for one OR MORE users; $139.50, and the second is a programmable 24 line parallel interface; $119.45. Both have ESE's full 64K selectable I/O address decoders.

ELECTRONIC SYSTEMS ENGINEERING
477 Congress Street, Suite 911
Portland, Maine 04104
MC/VISA (207) 773-7778

"MULTIFONT"

A NEW ADD-ON CARD FOR THE IBM PC/XT. "MULTIFONT" ENABLES THE IBM MONOCHROME PRINTER ADAPTER TO HAVE 256 USER DEFINABLE SYMBOLS. IT COMES WITH A POWERFUL CHARACTER/SYMBOL EDITOR, A PRINTING UTILITY, AND TEN ALREADY-MADE FONTS, INCLUDING APL, HEBREW, ARABIC, SCIENTIFIC, AND ELECTRONIC SYMBOLS, "MULTIFONT" IS TOTALLY TRANSPARENT TO THE SOFTWARE AND WORKS WITH ANY WORD PROCESSORS. LIST PRICE $249.95 HASH TECH CO., 2065 MARTIN AVE, #103, SANTA CLARA, CA 95050, (408) 988-2646.

INDUSTRIAL I/O

A full line of industrial I/O cards including RS-422, Four Port RS-232, and Combo Parallel/Serial Interface. Connect your PC to programmable controllers, data acquisition, keyboards, terminals, relay modules, or ADAMS NEW RS-422 REMOTE LCD-KEYBOARD MACHINE CONTROLLER. Other monitoring and control products available.

adams

ADAMS INC.
PO Box 17525
Greenville, SC 29606
(803) 297-9630

Hardware/Add On Boards

IEE-488 INTERFACE

Halfsize IEEE-488 interface and software turns your PC, AT or compatible into an instrumentation work station for the lab, test, and measurement. We are the major OEM supplier with interfaces for 10 different architectures and the most comprehensive coverage of operating systems and languages. $395 software $75.

NATIONAL INSTRUMENT
12109 Technology Blvd. Dept. PC
Austin, Texas 78727
(800) 531-5066, (512) 250-9119

Hardware/Disk Drives

1.2MB PC-AT DISK DRIVES

Replace your PC disk controller and full-height disk drive with our JDISKETTE controller plus our 1.2MB and 360k half-height drives for $499. Only $375 without 360K drive, $549 with two 1.2 MB drives. Put 800K on ordinary diskettes. 1.2MB diskettes are $59 per box (10). Mounting brackets, and JFORMAT included. Jdiskette also plugs into our JRAM-2 multifunction board.

TALL TREE SYSTEMS

TALL TREE SYSTEMS
1032 Elwell Court, Suite 124
Palo Alto, CA 94303
(415) 964-1980

Hardware/Peripherals

A/D 2 RS232 CONVERTER

Low cost stand-alone data acquisition board for any 8 bit RS 232 port. 8 channels, 8 bit conversion. 0-5V input, 300-9600 baud. Small size, 4" × 5", with wall plug power supply. Easy-to-use, includes sample driven software listings. Kit w/ all parts & PS, $59.95 + $4 shipping. Assembled & tested is $89.95 + $4.

TPS SYSTEMS
14820 Elmore Road
Anchorage, AK 99516
(907) 345-6730

Publications/Catalogs

TECH & USERS NEWSLETTER

Why read 20 magazines and books each month? The TECH LETTER summarizes the latest discoveries, problems, solutions, and news for hardware and software. Includes small programs and useful memory locations that are fully explained. $30/year. Also, Book/Software packages with 200 page manual and disk of examples/editors: #1 QUALITY GRAPHICS—$39.95; #2 GAME DESIGN—$39.95. SASE for catalog or order by check, VISA, M/C (add $3 ship).

COMPUTERS MADE SIMPLE
1974 Buck St.
Eugene, OR 97405
(503) 344-2767

dFASTER.dBETTER™

This performance book is for the serious dBASE user. Improvements of 20-60%, and even more are common. First use the insights, techniques and benchmarks. Then use dPROCESSOR™ to up speed even further. Everything you need to make dBASE II fly. $35 + $35H (Chk/MO/VISA/MC) or request info.

Micro db Systems
POB 2380 Dept. G
Midland, Michigan 48641

Software/Accounting

SOURCE CODE INCLUDED

Fully integrated innovative accounting, order processing, inventory, and sales analysis software. Available for a one time charge. Resell as often as you wish. NO ROYALTY PAYMENTS EVER! Written in C BASIC, in use since 1981 and over 1000 installed. Professional packaging and point of sale material available at cost. Includes 700 pages of documentation, Master Disks, and 10 sets of sales literature for $995.00.

TECHNOLOGY GROUP
2000 L Street N/W Suite 200
Washington, DC 20036
(416) 826-5320

Software/Business

TIME TRACK

Is a time recording and billing system that converts time spent by people within an organization into invoiced amounts. The system is menu driven for ease of operation. Online help screens supplement an extensive user guide. The license price of TIME TRACK is $495.00.

DOMUS Software Limited
251 Cooper Street
Ottawa, Canada K2P 0G2
1 (613) 230-6285

DATA SECURITY—$49.95

Secure sensitive data with DES FAST-CRYPT, an efficient Assembly language version of the powerful Data Encryption Standard algorithm. Encrypt/Decrypt any file at 120,000 bytes/min. Program or BASIC/PASCAL callable subroutines UNPROTECTED DISK w/manual—$49.95. VISA+MC OK. For IBM-PC, XT + compatibles 64K any PC/MS-DOS.

KIWI SOFTWARE CO.
P.O. Box 218 T
Plainsboro, NJ 08536
(718) 816-1873

RATES AND INFORMATION

Listings are grouped by category and consist of a bold lead line (23 characters maximum), 7 lines of ad copy (45 characters per line), plus 4 lines of company name, address and telephone number.

Listings are available only on a 3 issue basis at $90. per issue ($270. total). Copy will have a set format and remain the same for all 3 months. Enhance the appearance of your ad by including your Logo and an additional cost of $25. per issue ($75. minimum extra charge). Pre-payment is required by check, money order, or American Express, Diners Club, MasterCard, Visa credit cards. Closing Date: 1st of 2nd month preceding cover date.

Send copy and remittance to PC TECH JOURNAL, TECH BOOK, 12th FLOOR, 1 PARK AVENUE, NEW YORK, NY 10016. Call (212) 503-4506 for additional information or assistance.

TECH BOOK

IBM PC DATA ENTRY SYSTEM

The MARK 1 System allows fast and accurate Data Entry within a Job/Batch environment. The system includes: Entry, Verify, Search, Field Totals, Password Security, Field Edits, Custom Screen/Form Design, Comprehensive HELP Facility, Menu Driven and Much More. Files can be exported to PC DOS format for additional processing by the PC or for file transfer to other mini or mainframe computers. Runs on: PC, XT & AT. $295.

SUNUS SYSTEMS, INC.
4244 Ridge Lea Rd., Suite 25
Amherst, NY 14226
(716) 834-2820

SOFTWARE/BUSINESS OPPORTUNITIES

SOFTWARE AUTHORS

WE WILL CONNECT YOU TO MAJOR SOFTWARE PUBLISHERS FREE. We are in contact with many publishers looking for marketable software. If you have a quality software package or program and wish to have it published and distributed, contact:

SOFTSEARCH, Inc.
P.O. Box 281
Budd Lake, NJ 07828
(201) 627-1790

Attention: SOFTWARE DEVELOPERS

We are interested in receiving your completed package and business plan. Hammer Computer Systems, Inc. is in the software publishing business. Please send inquiries and complete portfolio to: Thomas A. Horvath, Vice President of Sales and Marketing.

HAMMER

HAMMER COMPUTER SYSTEMS, INC.
700 Larkspur Landing Circle/Suite 285
Larkspur, CA 94939
(415) 461-7633

SOFTWARE/COMMUNICATIONS

VT100 on PC, jr, XT AT

ZSTEMpc-VT100 Smart Terminal EMulator 132 columns by windowing—no addit. hardware. Double High/Double Wide Characters. Complete VT100 line graphics & Smooth scroll. Full keyboard softkeys/MACROS. Bidirectional file transfers including XMODEM. Speeds to 38.4KB. High throughput. $150. Quantity & Dealer Discounts Available. 30 day money back guarantee. MC/VISA accepted.

KEA SYSTEMS LTD.
#412-2150 W. Broadway
Vancouver, BC, CANADA V6K 4L9
Support (604) 732-7411—
Orders (800) 663-8702

SOFTWARE/COMPILER

DESMET C—$109

Full K&R C compiler, assembler, linker, librarian, full-screen editor and example software. Both 8087 and floating point libraries. OUTSTANDING PRICE/PERFORMANCE. Rated 1st or 2nd in August '83 BYTE benchmarks. No royalties on generated code. C Ware newsletter. Unlimited updates at $20 each. PC-DOS (Ver 1.1 & 2.0), generic MS-DOS and CP/M-86 support. $109 for complete package, shipping included. Now available with source level debugger. Price $159.

C WARE CORPORATION

C WARE CORPORATION—MAIL ORDER DEPT.
P.O. Box 710097
San Jose, CA 95171-0097

WIZARD C COMPILER

The power of C with the thorough diagnostics of PASCAL. Full LINT diagnostic integrated with compiler. Full UNIX SYSTEM 3 compatibility. Floating point library; fast compact code; 1 register variable. In-line assembly code. UNIX emulation library. Uses MICRO-SOFT linker. PC/MS DOS 2.0 only. $450.00.

WIZARD SYSTEMS SOFTWARE
11 Willow Court
Arlington, MA 02174
(617) 641-2379

SPARRY BASIC—B COMPILER

Powerful basic compiler that means business.
* Instructions support internal ISAM files.
* Four virtual screens, recall time <1 second.
* Use of all memory for basic programs.
* 13 significant digit integer math package.
* Demos with tutorial and reference manual. Price $159. Quantity discounts VISA/MC/CHECK. Mass. res. + 5% tax req. PC DOS with 128K.

SPARRY SOFTWARE LABS
PO Box 632 #A
Milford, MA 01757
(617) 473-5435

150 C FUNCTIONS $49.95

Functions, macros, assembly functions. Screen, cursor, keyboard, string, timing, BIOS and DOS, system status/control, utilities, random num. Many more you don't have but need! Just one may save five times cost of package! Get it in addition to/instead of any other library. 2 disks, all source code only $49.95. More to come: windows, screen mgr, graphics, execute.

Entelekon SOFTWARE SYSTEMS

Entelekon
12118 Kimberly
Houston, TX 77024
(713) 468-4412

COMPATIBLE LANGUAGES

C, Pascal and Multi-basic compilers generate compatible object code so languages can be mixed. C is full K&R, Pascal is full Jensen & Wirth + extensions and Multi-Basic compiles both MBASIC and CBASIC. $139 each. Available for PCDOS, MSDOS, CPM80; TRSDOS on Models I, II, III, 4, 12. Send for free brochures.

ALCOR SYSTEMS
13534 Preston Rd., Suite 365
Dallas, Texas 75240
(214) 494-1316

C SOFTWARE DEVELOPMENT

- Full C Compiler per K&R
  Inline 8087 or Assembler Floating Point
  Full 1 Mb Addressing for Code or Data
- MS DOS 1.1/2.0 Library Support
  Program Chaining Using Exec
  Environment Available to Main
- c-window™ C Source Code Debugger
- FAST 8088/8086 Assembler

COMBINED PACKAGE $199

c-systems
P.O. Box 3253
Fullerton, CA 92634
(714) 637-5362

SOFTWARE DEVELOPMENT TOOLS.

THE FORMS DESIGNER™

Attention IBM Pascal, FORTRAN users! Save time in designing formatted screen I/O. Interactive Forms Editor allows you to draw lines and boxes, define fields, and edit text. Access forms or read keyboard entry by writing only one line of code. Provides sequential data retrieval and storage. Requires 128K RAM. Only $275 complete. Demo and manual $35. Call or write:

BIT SOFTWARE
PO Box 619
Milipitas, CA 95035
(408) 262-1054

CBTREE for C PROGRAMMERS

Provides enhanced file access function calls directly into C programs. Maintains balanced B-trees, supports unlimited number of data records, keys, data files & key lengths. Dynamic memory allocation. Fast, Flexible, efficient. Object modules avail. or major C compilers. No royalties; Source code separate. $179.

MICRO COMPUTING SERVICES
2009 Hileman Road
Falls Church, VA 22043
(703) 893-0118

MATIS, The Screen Manager

70 Assembler subroutines used as BASIC commands, PROCEDURES in PASCAL, and CALLS in C and Assembler. Features: window management, easy screen design, control and saving to disk, printer utility, demo program. 170-page manual included. $150. CA add 6.5%. $2.50 shipping. No Royalty.

SOFTWAY
500 Sutter Street, Suite 222
San Francisco, CA 94102
(415) 397-4666

C-INDEX+

C-INDEX+ provides complete data management for C language applicatioins development.
- B+ Tree ISAM Indexing
- Variable Length Data Storage
- Full Data Integrity Protection
- High Performance and Flexibility
- Complete Source Code
- No Application Royalties

Fully supports Lattice and C86 compilers in large and small models. Available from Trio Systems and leading software retailers. One-time License Fee: $400. Demonstration: $25.

TRIO SYSTEMS
2210 Wilshire Blvd. Suite 289
Santa Monica, CA 90403
(213) 394-0796

TOOLS FOR TURBO PASCAL 40+ DEVELOPMENT SUBROUTINES

Including Debugging and Tuning Aids
SNAP HEX DUMP SORT
SET FKEY CMNDS DATE/TIME/DOW
ELAPSED TIMER SECTOR 1/D

PC DOS 2.0 TURBO PASCAL 2.0 Source incl. Send $29.95 check/MC/VISA (TX pay $31.78)

ALLEN, EMERSON & FRANKLIN
PO BOX 928
Katy, TX 77449
(713) 391-8570

PROGRAM DEVELOPMENT ENVIRONMENT FOR BASIC

A full screen editor designed for the BASIC language. Automatic update of line numbers for all edit functions. Group move, copy, delete. Global search, replace. Programmable keyboard, macros, undefined line number check, much more. SINGLE KEY SWITCH BETWEEN THE EDITOR AND BASIC INTERPRETER. Program is directly transferred from the editor to BASIC's workspace and back. Single key invocation of IBM BASIC compiler. Requires IBM-PC or XT, DOS 1.1 thru 2.1, 192k. $39.95 Money back guarantee.

JACKSON SOFTWARE
1729 Mayflower Drive
Carrollton, TX 75007

ASSEMBLE 3-6 TIMES FASTER

FAST ASSEM-86 is much faster and easier to use than the IBM Assembler. Assembles 8K files 6 times faster. Editor, assembler and source always in RAM so you can go instantly from editing to assembling. No need to LINK since produces .COM (as well as .OBJ) files directly. Source compatible with IBM's important features. Supports 186, 286, 8087, 287 directly. X-ref. utility built in. Only $199.

SPEEDWARE
Box T1, 2931 Northrop Avenue
Sacramento, CA 95825
(916) 966-6247

RAPID COMPILER DESIGN

An LR(1) parser generator complete with sample compilers. QPARSER creates a parser, a lexical analyzer and skeleton semantics with universal error recovery & interactive debugging. Adaptable to other languages. Runs on IBM PC. Textbook available in 1985. Educational/quantity discounts. FREE BROCHURE. $1995. from QCAD SYSTEMS, INC.
P.O. Box 700129, 1164 Hyde Ave.
San Jose, CA 95129
TOLL FREE (800) 538-9787;
(408) 255-5574 in CA

TECH BOOK

2/3D TOOLS for DeSmet C
Create dynamic symbol and image displays with a high performance graphics toolkit. FAST; to 290K pixels a sec., TIGHT; 6K. COMPLETE; invoke line, block, pattern, clip, text, windows, popups hi-res, color, and loon, Editor on PC, XT, AT color card. Order, call or write for more info. $55.
AFFINITY INC.
126 Normal Ave.
Buffalo, NY 14213
(716) 882-5077

See AN EDITOR BEST BUY!
See is for programmers. Fastest screens. Fastest load & ready. Easy to learn with On-Screen help, tutorial. 20 active file-windows 256 ch. lines. Char.-line, block & file do undo operations. Auto-indent, Wildcard search-replace. No file size limits. Invoke DOS, run a program, return as was. Full-features. Need: PC, jr, XT or AT; 128K; DOS 2.0+. Inquire UNIX version. $69.50 demo $12.50.
PROLOGIC CORPORATION
31324 Via Colinas, Suite 111
Westlake Village, CA 91362
MC/VISA (818) 991-5062

SYMBOLIC DEBUGGER
Save development time with the WATCHDOG SYMBOLIC DEBUGGER. Features: Two full screen debug displays; Protected user screen; Labels for display and address entry; Command menu and prompts; Page and line scrolling; inline assembly; Repeat count for Breakpoint and Step; Plus much more. DOS 1.x, 2.x, IBM PC/XT, 128K. Introductory price $129.
SOUTHERN CROSS SOFTWARE, INC.
PO BOX 3016
BURNSVILLE, MN 55337
(612) 890-8738

MOUSE + WINDOW EDITOR $50
Fast moused-based editor with overlapped windows allows effective use of the display. All commands given with mouse: move, copy, select, etc. for text; create, destroy, zoom (to whole screen and back), scroll, change size, move, top (& more) for windows. Similar to WORD (but no formatting). $49.95 + $3 shipping.
LEMMA SYSTEMS
11355 Missouri Ave.
Los Angeles, CA 90025
(213) 473-0171

IMPROVED DOS SHELL $40
Programmer's shell: select all or part of previous commands; select text with a mouse; expand file name patterns in-line; flexible command line cursor motion; insert and delete anywhere; fast, wide, sorted version of DIR; fast, flexible file pager; redefine keys; CD with menu; and more. $39.95 + $3 shipping.
LEMMA SYSTEMS
11355 Missouri Ave.
Los Angeles, CA 90025
(213) 473-0171

OMNISCREEN
Complete SCREEN MANAGEMENT. The FORMATTER provides screen Creation/Testing/Documentation. Color, all video attributes, data input edits, boxes. The PROCESSOR has over 30 calls to control data entry & display. Ins/Del & all cursor movement keys & 40 function keys. PASCAL/ASM/COBOL/C/FORTRAN/BASIC. $250.
OMNISOFT ASSOCIATES
6917 12th Ave.
Brooklyn, NY 11228
(718) 680-3259

THE HAMMER—tools in C
More than just BOLD/DOS access, THE HAMMER ALSO provides screen attribute control, Smart modem-control, UNIX-like pattern matching, 123-like command menus, easy data entry & verification of dates, numbers and string AND MORE. Routines tried & true. $110.00 with source. Call or write now:
OES SYSTEMS
1906 Brushcliff Road
Pittsburgh, PA 15221
(412) 244-1086

BTREES IN C WITH SOURCE
The Softfocus B-Tree Library is a complete set of data management functions written in K&R C. We support random and sequential access, multiple indicies per data file and up to 16.7 million keys per index. Also, our software is customizable because we supply full source code. Manual & examples included.

B-Trees
$77.00

SOFTFOCUS
1277 Pallatine Drive
Oakville, ONT. CANADA L6H 1Z1
(416) 844-2610

SOFTWARE/ENGINEERING

PC ENHANCEMENT HANDBOOK
THE IBM PC ENHANCEMENT HANDBOOK for SCIENTISTS & ENGINEERS. A How-To-Handbook with answers to all of your questions on PC enhancement. What can be done? How to do it? and How much will it cost? Includes Toll Free Hotline for application assistance & convenient one stop shopping. SPECIAL INTRODUCTORY OFFER $18.95.
CyberResearch, Inc.
P.O. Box 9565
New Haven, CT 06536
(800) 341-2525; (203) 436-2600 (CT)

smARTWORK™
Printed-circuit artwork editor for double-sided boards up to 10" by 16", runs on an IBM-PC. Color display allows complete interactive control over the placement and routing process. 2X artwork can be made on a dot-matrix printer or pen-and-ink plotter, $895. Write or call for a brochure.

WINTEK

WINTEK Corporation
1801 South Street
Lafayette, IN 47904-2993
(317) 742-8428

SOFTWARE/FILE MANAGEMENT

A FRIENDLY FILER
There has NEVER been a filing program easier to use than SeekEasy. NEVER! It uses an advanced "flexible match" algorithm. Just type APPROXIMATELY what you want to find seeking easy does the rest. Fast! PC/MS-DOS 2.0. $87 (demo $15)—add 6.5% in CA. Add $3 S&H. VISA/MC O.K. Free brochure & reviews.
CORRELATION SYSTEMS
81 Rockinghorse Road
Rancho Palos Verdes, CA 90274
(213) 833-3462

SOFTWARE/FINANCIAL

RORY TYCOON
Portfolio Manager for small investors with big ideas and big investors with small computers. Track stocks, options, CDs, IRAs, cash, bonds (zeros also), metals, mutual funds, and profit or loss, yield, asset distribution, etc. $25.00 (Texas + tax). Requires Lotus 123.
COHERENT SOFTWARE SYSTEMS
311 W. 21st St.
Houston, TX 77008
(713) 869-0465

SOFTWARE GENERAL

RENT PUBLIC DOMAIN SOFT/W
Rent our PC Disk Libraries for 7 days and copy them yourself. Hundreds of useful business, games and utilities FREE FOR THE TAKING!

IBM-PC SIG-135 Disks	$135.00
IBM-PC "Blue" 53 Disks	$50.00

Rental is for 7 days after receipt, 3 days grace to return. Most credit cards accepted.
NATIONAL PUBLIC DOMAIN
SOFTWARE CENTER
1533 Avohill Drive
Vista, CA 92083
(619) 941-0925. Info-(619) 727-1015.
24 hr. orders.

ATTENTION SOFTWARE AUTHORS
Our established literary agency is seeking to represent talented freelance programmers. We open doors for our clients at the leading software houses and negotiate the best deal possible. Put our years of experience to work for you. For further information on the benefits of representation contact:
THE ROBERT JACOB AGENCY
1642 Eveningside Drive, Suite 110
Thousand Oaks, CA 91362
(805) 492-3597

SOFTWARE/GRAPHICS

GEOGRAF™
GEOGRAF™ helps you prepare high-quality, customized graphs for business, engineering and science. GEOGRAPH™ cuts programming time by up to 80%. Callable from BASIC, FORTRAN or PASCAL. Versions available for most graphics cards and plotters. Fully compatible with Colcomp's FORTRAN standard plotting package.
GEOCOMP CORPORATION
342 Sudbury Road
Concord, MA 01742
(617) 369-8304

PRESS 'N' PLOT 2.0
Insert images into your text for correspondence that counts. Capture screen images from most graphics programs and merge them with text from most word processors. Can automatically wrap the text around the image. Supports most popular color or B-W dot matrix printers. $149.
AMERICAN PROGRAMMERS GUILD, LTD.
55 Mill Plain 17-5
Danbury, CT 06811
(800) 828-8088; in CT, (203) 794-0396

TEKTRONIX 4010 EMULATION
High resolution screen/printer graphics for the IBM PC. Full interactive capabilities with file transfer and cross-hair control. Utilizes PLOT 10, protocols. OFF-line review of graphics output. Supports IBM color/graphic or Hercules cards. Easy to use 4010 emulation at an affordable price. $80.00. Demo disk $5.
Technological Systems Group
5044 Haley Court
Lilburn, GA 30247
(404) 923-4980

GRAPHICS LIB FOR TECMAR
TEK-MAR lets you do high-res graphics on your TECMAR Graphics Master. TEK-MAR is a library for use with MS Fortran. Features windowing, viewporting, clipping, axis rotation. Similar to Tektronix graphics. Includes screen dump/restore, Epson screen print, support for (Western Graphtec) plotter. Requires MS-FORT 3.20, 320K, GMDEV.SYS (5.2). Price: $195.
ADVANCED SYSTEMS CONSULTANTS
18653 Venture Boulevard, Suite 351
Tarzana, California 91356
(818) 990-4942

SOFTWARE/LANGUAGES

SNOBOL4+
Mainframe SNOBOL4 on your desktop, plus binary and random-access I/O, SAVE files, & more! Unexcelled pattern-matching and string manipulation, integer & real math, link to assembly language. Includes ELIZA and over 100 sample programs and functions. 128K to 448K, DOS or CP/M-85. Only $95 + $3 shipping.
CATSPAW, INC.
P.O. Box 1123
Salida, CO 81201
(303) 539-3884

TECH BOOK

Software/Operating Systems

MULTIPLE USERS UNDER PC DOS

MultiLink turns PC-DOS into an efficient multi-user multi-tasking operating system. Additional users are supported by attaching inexpensive CRT terminals or modems to serial ports on the PC, and can run normal applications designed for PC-DOS. Includes host communications software for public dial-in.
THE SOFTWARE LINK INCORPORATED
6700 23B Roswell Rd.
Atlanta, GA 30328
(404) 255-1254

Software/Programming Tools

W-ED, a small, fast and efficient editor/word-processor can be embedded into BASICA, compiled BASIC, IBM assembler, and selected C and Pascal language programs on the PC, AT, jr., and compatibles. File paths and windowing are supported. Programmable version(s)—$140/ea. Stand-alone—$40. Demo $6. Brochure. MC/VISA
WyndhamWare
PO Box 729
Renton, WA 98057 0729

Software/Services

TAPE TO DISK CONVERSIONS

Convert any 9 track magnetic tape to or from over 200 formats including 3½", 5¼", 8" disk formats and word processors. Formats available include IBM-PC, Apple, Altos, TRS 80 8" CP/M, Displayer., IBM Sys/??. Macintosh, Wang, Lanier, OS/6 and 200 more. Disk to Disk conversions also avail. Call for more info.
PIVAR COMPUTING SERVICES, INC.
47 W. Dundee Rd.
Wheeling, IL 60090
(312) 459-6010

Software/Sort

OPT-TECH SORT/MERGE

Extremely fast Sort/Merge program for the IBM-PC. Can sort or merge multiple files containing fixed or variable length records. Run as a DOS command or call as a subroutine, plus many other features. Now also sorts dBASE II files! Compare before you buy any other. Write or call for more info. $99.
OPT-TECH DATA PROCESSING
P.O. Box 2167
Humble, TX 77347
(713) 454-7428

INTERNAL SORT FOR BASIC

NSORT—A FAST & easy to use quicksort subroutine written in C & assembler, CALLable from BASIC. Sorts an unlimited number of records in memory on up to 8 keys of mixed type, ascending/descending. Like having a SORT built into the language. Works w/compatibles, Compiled and Interpreted, all DOS versions. $49.95 + $2.50 s&h. MC/VISA/CHECK. PA + 6% tax.
NATIONAL SOFTWORKS
65 East Elizabeth Avenue
Bethlehem, PA 18018
(215) 867-4800

Software/Taxes

MULTIPLAN TAX TEMPLATES

Computes the LOWEST POSSIBLE TAX & saves time and frustration. 24 forms and schedules-in an IRS-acceptable format. Fast, modifiable and expandable. Use all year, Annual updates (or update yourself). Requires Multiplan. $89 + $4.50 S/H. Multiplan, $95: BOTH $174. We have other products. MC/VISA. ORDER NOW!
VISION INFORMATION PRODUCTS, INC.
5500 Atherton Street, Suite 306
Long Beach, CA 90815
(213) 431-5284

1-2-3 TAX TEMPLATE

Worksheet calculates and carries entries across Forms 1040, A,B,C,D,E,SE,W, 2106, 2441, 3468, and 4562. Ledger feature allows tax planning all year. $25.00 (Texas + tax). Requires Lotus 1-2-3.
COHERENT SOFTWARE SYSTEMS
311 W. 21st St.
Houston, TX 77008
(713) 869-0465

Software/Terminal

BLUELYNX

Replace your dumb terminals with smart PCs and XTs. TECHLAND SYSTEMS is the nation's leading synchronous communications company with over 6,000 BLUELYNX' installed. The BLUE-LYNX hardware/software product emulates the 5251/12 for the S/34, S/36 and S/38. Now bundled with Autolynx for unattended operation ... only $745. BLUELYNX 3276 for the larger 43xx and 30xx mainframes, both SDLC and BSC versions, now comes bundled with Data Reader, a utility that facilitates the formatting of host files into Lotus 1-2-3, dBase, etc. ... only $795. Printer and up-and-download support available with both versions.

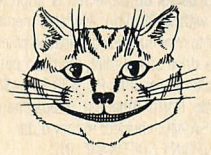
TECHLAND SYSTEMS INC.

TECHLAND SYSTEMS, INC.
25 Waterside Plaza
New York, NY 10010
(212) 684-7788

Software/Terminal Emulation

VT52/VT100/TEKTRONIX 4010

ZAP emulates the above terminals and includes nested environment files, XMODEM & Kermit file transfer, auto-redial, key bindings, integrated text & graphics, save/restore graphics utility and dump mode. XON/XOFF. Create variable-size graphics characters. Requires DOS 128K. $65+ $5 shipping (PA residents add 6%)



CHESHIRE CAT SOFTWARE
PO Box S
Devon, PA 19333
(215) 644-4914

Software/Typesetting

HIGH-TECH TYPESETTING

Transmit your text via toll-free lines directly to our fully automated typesetting system. $2. per K characters with a $5. minimum. Same day service. 200 typefaces in sizes up to 72 point. Send $15. + $3. shipping for our 220 page guidebook, or call toll free and use your MC, VISA or AMEX.
INTERGRAPHICS INC.
106A South Columbus Street
Alexandria, VA 22314
(800) 368-3342 or (703) 683-9414 in DC area.

Software/Utilities

CONVERTS ANY CPM TO DOS

CROSSDATA CONVERTS ANY DATA/TEXT FILE format from CPMxx to MS/PC-DOS, CPMxx to CPMxx & MS/PC-DOS to CPMxx. Cross data runs on IBM PC or comparable computer using MS/DOS 2.0. Cross data is a self-contained program. It comes with over 24 mats and user can add own format. To order send $99. check or money order to:

AWARD SOFTWARE, INC.

AWARD SOFTWARE, INC.
236 North Santa Cruz Ave.
Los Gatos, CA 95030
(408) 395-2773

Hard Disk DIRECT ACCESS™

The ULTIMATE utility for any hard disk user. DIRECT ACCESS is a powerful hard disk management tool. Organizes your software programs into a "user defined" menu system. Features attractive screen display, rapid access into an application using a single keystroke & easy to learn. Also handles batch files & complicated DOS commands. Order today. Only $59.95 + $2.50 shipping & handling. MC/VISA.

DELTA TECHNOLOGY

DELTA TECHNOLOGY
P.O. Box 1104
Eau Claire, Wisconsin 54702
(715) 832-0958

PADLOCK/PADLOCK II DISKS

PADLOCK furnishes the user with a method for providing protection against unauthorized duplication from DOS commands $99. PADLOCK II disks come preformatted with finger-print and serialization. PADLOCK II disks offer superior protection. Ask about our fast data encryption product. All work with hard disk, EXE/COM files and all DOS versions. MC/VISA.
GLENCO ENGINEERING
3920 Ridge Ave.
Arlington Hts., IL 60004
(312) 392-2492

FULL CONTROL FROM BASIC

Cut development time, add speed & power to your programs w/NLIB. A library of assembly language subroutines callable from BASIC: execute COM & EXE files, save & restore areas of the screen, clear & scroll windows, much more! Add DOS functions to your menus. Create fast help screens. Unlimited uses. All DOS Versions $49.95 + $2.50 s&h. MC/VISA/CHECK. PA + 6% tax.
NATIONAL SOFTWORKS
65 East Elizabeth Ave.
Bethlehem, PA 18018
(215) 867-4800

SCREEN DESIGN MADE EASY!!

Screen Generator and Processor for BASIC programmers. This package combines a Screen Painter for creating and modifying screens and Macro Language and Compiler for quicker coding and testing. Features COLOR, field variables, complete input editing, line drawing. For BASIC, BASICA, and BASIC Compiler users. Includes manual and reference card. $99.95 VISA/MC accepted.

BasicWindow™

G. FREEMAN & COMPANY, INC.
15 Albin Road
Stamford, CT 06902
(203) 327-9868

TECH BOOK

DOS PATH COMMAND FOR DATA

Now Lotus 1-2-3, dBASE II, WordStar and most others can access data files no matter where they are located! Works with hard disks and floppies. DPATH is a system stay-resident function that allows you to organize your disk any way you like. Includes a screen-oriented maintenance utility & a 60-page manual. DOS 2.0/compat. $25.
PERSONAL BUSINESS SOLUTIONS INC.
PO BOX 757
Frederick, MD 21701
(301) 865-3376

DISK MECHANIC

THE ULTIMATE Floppy Disk Backup & Repair Utility. Can back up ALL software protected disks written on the IBM PC. Works manually or automatically. Files or sectors can be restored, searched, examined & changed. Checks disk drive speed. req. IBM PC, DOS 1.1 (128K)/2 (192K) + 64K if only 1 floppy drive. $73 ppd. USA
MLI MICROSYSTEMS
PO BOX 825, Dept. TB
Framingham, MA 01701 USA
(617) 926-2055 for info MC/VISA

CopyWrite

CopyWrite backs up IBM PC software. We have not found anything that we can't copy. CopyWrite is revised monthly to keep up with the latest in copy protection, and comes with a trade-in offer. It needs an IBM PC, 128K and one disk drive, but can use more memory or another drive. $50 US. Check/Credit Cards. U.S. Inquiries.
QUAID SOFTWARE LIMITED
45 Charles St. East
6th Flr.
Toronto, Ontario M5Y 1S2 Canada

COPY PROTECTION

SLK/F places an assembled or compiled program on a diskette with 4 different copy-resistant features in such a way that it runs normally, but cannot be copied by backup programs such as COPYPC. The rest of the diskette is available as normal, and DOS may be added. Price $150.

OLIVE BRANCH SOFTWARE

OLIVE BRANCH SOFTWARE
1715 Olive Street
Santa Barbara, CA 93101
(805) 569-1682

SS/MS/IBM FORTRAN USERS

Utility subroutines that includes DOS/BIOS gates, cursor, screen, inkey, free read of text or numbers with function key control, output scrolling, line and window clear, capset, numlock, music, draws rectangles, statistical and financial routines. $40 IBM 2.0/MS 3.2, $45 SuperSoft 1.0 or 2.0 (Demo $8 w/freeware financial program). Chk./M.O.
COMPUTER HANDHOLDERS
P.O. Box 59
Arcola, PA 19420
(215) 565-7467

WHERE ARE THEY NOW?

Have you ever forgotten which directory a file is in? The WHERE program searches through all directories for the file you ask for, displaying the path of each matching file name. WHERE is easy to use, accepts? and *, and is indispensable for clean up and reorganization. WHERE requires MS or PC DOS 2.00 and above. $20 (includes s&h).
KING PROGRAMMING PRODUCTS
P.O. Box 432
Wood Dale, IL 60191
(312) 350-0100

COPY II PC

CPY II PC backs up protected & unprotected software, with verification. Available memory is full utilized to minimize disk duplication time. Also includes a drive speed utility that can pinpoint drive problems. For IBM-PC. $39.95 + $3 s/h ($8 outside North America). Prepayment required: VISA/MC, check or money order.
CENTRAL POINT SOFTWARE, INC.
9700 SW Capitol Hwy., #100T
Portland, OR 97219
(503) 244-5782

PC ALARM CLOCK FOR $30

STOP MISSING IMPORTANT APPOINTMENTS. PC ALARM provides an alarm function and time-of-day display for the IBM-PC/XT under DOS (1.0 or later). Initiate ALARM with times & messages, then forget it as you run other software, until ALARM signals your attention with both audible and visual displays. Mono or RGB.
LAF SOFTWARE
3 Kiowa Lane
Los Alamos, NM 87544
(505) 672-3359

PRINT SCREEN

Prints reverse video, underlining, high and low intensities and all 256 characters for IBM PC text screens. Graphic screens can be printed in a variety of sizes, vertically or horizontally. A variety of printers are supported. The license price for PRINT SCREEN is $50.00.

DOMUS SOFTWARE LIMITED
251 Cooper Street
Ottawa, Ontario K2P OG2
1-(613) 230-6285

PERSONAL CHARACTERS

Create character sets, print IBM, foreign & Greek characters, change printer modes & fonts using DOS, Alter character sets, download characters sets using DOS and BASIC programs, printer driver for microsoft "word", Epson FX-LQ-1500, or Okidata 9× printers. Minimum requirements: 128K, & DOS 2.0. MC/VISA/CHECK $49.50.
GROFAR SYSTEMS
26 W. Faculty Rd.
Thousand Oaks, CA 91360

END DISK INCOMPATIBILITY

with CONVERT—the critically acclaimed disk/conversion program. CONVERT runs on the IBM/PC and compatibles. It reads, writes and FORMATS diskettes for over 40 different CP/M computers. CONVERT is the standard disk conversion software. Buy the best for only $99. VISA/MC accepted.
SELFWARE, INC.
3545 Chain Bridge Rd., Suite 3
Fairfax, VA 22030
(800) 242-4355

BASIC "QUICK-SCREEN"

Create complex data input screens fast and easy. Place "data fields" anywhere you want. Draw menues and help screens. Comes with subroutines to manage screens, data input, etc. Includes a program listing utility to demonstrate "QUICK-SCREEN". Works with compiler. $38 + shipping, $8 for demo. CA res. +6% tax.
RIBB SOFTWARE
20 Quiet Hills Road
Pomona, CA 91766
(714) 622-6653

ELIMINATE IO ERROR = 98.

If you lose time to IO ERROR = 98 in your RM/COBOL applications, VERIFILE can save you hours of restructuring time. VERIFILE is a DOS command file that detects and corrects errors in RM/COBOL ISAM files before programs are executed. Disk and reference manual included. Send check or money order for $69.96 to:
GREYSTONE GROUP, INC.
214 Second Avenue North
Nashville, TN 37201
(615) 248-9119

FILE HONEY

No more memorizing subdirectory names, waiting for "Dir" or "Tree" commands, or fumbling to identify files. This directory-scanning aid uses windows for rapid access to your subdirectory tree; allows scrolling of sorted file directories; and much more. For PC/XT with 128K & DOS 2.0. Price $42.50—demo $5.00.
JOE PHILLIPS CO.
9071 Metcalf, Suite 145-C
Shawnee Mission, KS 66212
(913) 341-3645

DIGICON PRINT PACKAGE

Want SIDEWAYS printing for SPREADSHEETS? How about BANNNER, PAMPHLET, LETTER, and BOOKLET printing too? Variety of fonts. Graphics characters. Character editor to MAKE YOUR OWN FONTS or SYMBOLS! Quick printer setup program too. Easy menus, online help. IBM/Epson printers, DOS, 128K. $49.95 (PA+6%).
DIGITAL CONCEPTS, INC
POB 8345
Pittsburgh, PA 15218
(412) 823-8314

A FRIEND FOR TURBO PASCAL

PASCAL'S FRIEND is four include files giving easy BIO/DOS access with examples plus easier use of Blaise Tools/MS Pascal code, keyboard, cursor, and screen handling, inverse video, system clock, calendar calculations, 1-2-3 like menu routines. Source for example program. $39.95 check, MO, Visa, MC.
J.S. COMPUTING
434 N. Lumber St.
Allentown, PA 18102
(215) 821-9020

SPEED UP YOUR DISPLAY!

FANSI-CONSOLE™: faster screen writing thru PC-DOS or BIOS (1.3-3.0+ times), larger X3.64 subset than ANSI.SYS, VT100/52 keys, more type-ahead, auto inactivity screen disable, eliminate scroll blink for some adaptors, window support, keyboard breakpts. Software & printed manual $50, or shareware disk $25.
HERSHEY MICRO CONSULTING, INC.
P.O. Box 8276
Ann Arbor, Michigan 48107
(313) 994-3259

SOFTWARE/WORD PROCESSING

VOLKSWRITER DELUXE

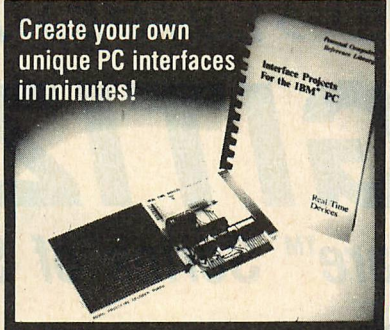
Features TextMerge™ for customized mailing lists; multi-lingual, engineering & scientific symbols; horizontal scrolling to 250 characters; document size 1 megabyte; on-screen page endings, underline, boldface, double-strike, strikethrough, multiple fonts & colors—fewest keystrokes per function. Only $295.
LIFETREE SOFTWARE INC.
411 Pacific Street
Monterey, CA 93940-2790

TECH MART

FORTRAN
or
PASCAL
PROGRAMMER?

READ
THIS
AD

Create your own unique PC interfaces in minutes!



The PD100 allows rapid development of specialized PC interfaces. It features a buffered data bus, switch-selectable address decoder, and a large prototype area (up to 40 IC sockets). The 116-page manual covers basic interfacing concepts and details implementing A/D, D/A converters, I/O ports, connection of transducers and dozens of useful circuits.

PD100 w/manual - $99 plus $3.50 P&H
Manual only - $20 Postpaid
PA residents add 6%
Check, MasterCard, or VISA

REAL TIME DEVICES
1930 Park Forest Ave.
P.O. Box 906
State College, PA 16804
(814) 234-8087

Cables
EIA RS 232-C

Send for Catalog

Quality cables with immediate delivery and low prices.

Conductor	Price
1-4	11.50 + .15/ft.
5-7	12.00 + .22/ft.
8-12	13.00 + .30/ft.
13-16	14.00 + .40/ft.
17-25	16.00 + .50/ft.

Specify Male or female connectors, length of cable and pins to be connected. Extended Distance, Centronics (Parallel), Coaxial (RG59U, 62A/U), Dual Wang, Twin-axial, Ribbon, IBM, DEC Compatible cables and AB Switches also available.

We supply connector parts, bulk cable, tools and hardware (wall plates), for those who prefer to build their own cables.

Communication Cable Company
PO Box 600-K, Wayne, PA 19087
215-964-9404

CIRCLE 303 ON READER SERVICE CARD

GRAPHICS
- Text/graphics generics
- 2D interactive
- 2D plots (full support) including auto-scaling, auto-axis generation, auto-labeling, tabular/log/parametric curves, splines, bars, pies, you name it, we have it.
- 3D Plots incl. 2 hidden line removal options—not just old-fashioned wire frame.

PEN PLOTS
- Standard plotter primitives plus
- FULL 2D support plus
- Interface to screen graphics,
- Limited 3D.

Clear and complete documentation.

GRAFMATIC $135.
PLOTMATIC 135.

MICROCOMPATIBLES
11443 Oak Leaf Drive
Silver Spring, MD 20901
(301) 593-0683

CIRCLE 305 ON READER SERVICE CARD

So You Love Your Work Keep It Fun
Let our fast stand alone

Cobol Cross Reference

take the drudgery out of programming!

Your IBM PC w/128K, DOS 2.0, one disk drive and our Program;

- **Will** cross reference and/or print Source Code
- **Will** flag duplicate data names and invalid references.
- **Will** allow more than 1400 data names and 11,000 references
- **Will** process all versions of Cobol
- **Will** be "personalized" with your name on the report heading.

Invest $25.00 in your future

Send check or money order to:

Meta System Inc. of Alaska
2806 Iris Drive
Anchorage, Alaska 99503
Phone 907-243-8619

CIRCLE 306 ON READER SERVICE CARD

NEW BOBCAT 3 DISK CATALOG

The most versatile and powerful catalog program available

- completely rewritten in C
- creates, adds, updates, and deletes a filename catalog
- seven report formats
- hard disks, multiple drives, and user numbers
- individual file titles
- wildcard searches for filenames and file titles
- CP/M or PC/MS DOS

8" CP/M SSSD or Popular CP/M or PC/MS DOS 5¼"

U.S. residents .....$49.95 U.S.$
Canadian residents ...$49.95 Cdn $ (Ont. residents add $3.50 pst)
Other countries .....$54.95 U.S.$
plus $3.00 P & H

NAME, NUMBER
EFFECTIVE & EXPIRY DATES
(MC FOUR EXTRA DIGITS)

Bank drafts; certified checks; money orders; company checks

R&L Micro Consulting Services
Box 15955, Station F
Ottawa, Ont.
K2C 3S8 (613) 225-7904 THE HOME OF THE BOBCAT

*MICROGUIDE is a ™ of R&L Micro Consulting Services
CP/M is a ™ of Digital Research.

CIRCLE 302 ON READER SERVICE CARD

ICs PROMPT DELIVERY!!!
SAME DAY SHIPPING (USUALLY)

8087-3	Co-Processors		$124.97
DYNAMIC RAM			
256K	256Kx1	120 ns	$19.97
256K	256Kx1	150 ns	$14.99
64K	64Kx1	150 ns	2.75
64K	64Kx1	200 ns	3.07
EPROM			
27256	32Kx8	300 ns	$36.25
27128	16Kx8	250 ns	13.57
27C64	8Kx8	200 ns	12.47
2764	8Kx8	250 ns	5.29
2732A	4Kx8	250 ns	6.37
2716	2Kx8	450 ns	3.31
STATIC RAM			
6264P-15	8Kx8	150 ns	$19.57
6116P-3	2Kx8	150 ns	4.37

QUANTITY ONE PRICES SHOWN

Open 6½ days: We can ship via Fed-Ex on Sat.

MasterCard/VISA or UPS CASH COD
Factory New, Prime Parts µP∞
MICROPROCESSORS UNLIMITED
24,000 South Peoria Ave.
BEGGS, OK. 74421 **(918) 267-4961**
Prices shown above are for December 17, 1984
Please call for current prices & volume discount. Prices subject to change. Please expect higher or lower prices on some parts due to supply & demand and our changing costs. Shipping & insurance extra. Cash discount prices shown. Small orders received by 6 PM CST can usually be delivered to you by the next morning, via Federal Express Standard Air @ $6.75!

RTL
Relocatable Threaded Language

RTL is a threaded language similar to FORTH but is based on a new type of threaded code called status threaded code. This new type of threaded code allows the inclusion of a number of features not normally included in threaded languages. Each word is independently relocatable. RTL allows editing of any high level word directly from RAM, eliminating the need for screens. Any word may be redefined retroactively so there is no kernel of unmodifyable words. RTL is now available for the 68000, 6809, 8086, and Z80.

RTL Programming Aids **$150**
10844 Deerwood SE
Lowell, MI 49331 **Includes all**
(616) 897-5672 **source code.**

CIRCLE 304 ON READER SERVICE CARD

PC THE BUYERS GUIDE

THE MASTER DIRECTORY OF PRODUCTS FOR THE IBM PC, PC XT, PCjr AND MOST COMPATIBLES!

AVAILABLE AT YOUR LOCAL NEWSSTAND AND COMPUTER STORE.

Don't SYNTHESIZE

DIGITIZE!

SoundWare™ Series of Software

Sound Application Development Tools...

Digital Pathways' SoundTools™ software used with the Communicard™ allows application software to use digitized voice messages, greetings, errors and much, much more. Touch Tone™ decoding lets applications software use the Touch Tone phone or pad as an input device for the remote or stationary user. Touch Tone dialing allows your PC application software to address such areas as telemarketing and auto dial applications.

Think of the Possibilities!

- *Sales order entry*
- *Order acknowledgement*
- *Message broadcasting*
- *Telephone management*
- *Voice distribution*
- *Computer aided instruction*
- *Telemarketing*
- *Dictation*
- *Order/quantity inquiry*
- *Text to speech*
- *Voice mail*
- *Inventory Adjustment*
- *Phone-in surveys*
- *Dial-in newsletter*
- *Text & speech dictation*

Languages: *Interpreted and compiled IBM™ BASICA and MS™ BASIC, MS PASCAL, DeSmet C and Assembler.*

Communicard*—This ½ size card provides complete telephone interface, microphone and auxiliary output and TouchTone decoding.*

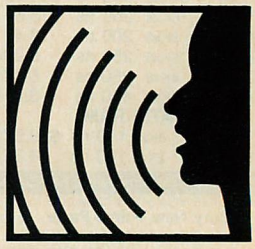
VoiceMate™*—Intelligent telephone management software for your personal computer. Includes phone directory, auto dialing, dictation, voice file transmission, audit trail, security, voice mail box and remote access.*

List price $449.00. Introductory price $360.00 until 2/28/85.
Includes Communicard, SoundTools and VoiceMate software, telephone cord and user's manual. Ask about 21 day trial offer when placing your order.

Requirements: *PC/MS-DOS 2.0 or 2.1, IBM PC/XT or compatible, 192K memory, DMA channel 1.*

You too, can put your software on a SOUND FOOTING...

SoundWare™
SERIES OF SOFTWARE



DIGITAL PATHWAYS, INC.
1060 EAST MEADOW CIRCLE, PALO ALTO, CA 94303, 415-493-5544

PC TECH JOURNAL PRODUCT INDEX

What do IBM, HP, Tektronix, and DEC have in common?

TBASIC!

Terrific TBASIC

A new, portable, incrementally compiled language with integral GPIB and Graphic statements compatible with Tektronix 4050A BASIC.

Graphics Galore

TBASIC gives you over 60 high level graphic commands for a variety of graphic computers and terminals. From IBM PC to high resolution Tektronix terminals, TBASIC lets you access their intelligent features consistently and with a minimum of effort.

GPIB Genius

Instrument control and operation of shared peripherals does not need to be awkward. TBASIC provides syntax for both simple access of GPIB devices through normal I/O commands that understand GPIB, and special purpose functions for sophisticated instrument control.

For Professionals and Part-timers

TBASIC is an excellent development language for general purpose or technical programs. It is extremely easy to learn and use with features that professional and novice programmers alike will find useful for applications ranging from business to scientific.

TBASIC Triumphs

No other high level language gives you complete access to and control over all elements of your computer system like TBASIC does. No other language gives you so many advanced and useful features.

Fantastic Features

- Graphics: over 60 statements
- GPIB access is incorporated into standard I/O statements
- Tektronix 4050A BASIC programs are compatible with TBASIC's superset
- Multiple statements per line
- Automatic line numbering
- Instant syntax checking "shows" errors
- Line editor with 7 function keys
- 31 character variable names
- Rename variables in programs with special edit command
- Cross-references lines and variables
- 20 user definable function keys
- Line labels for structured programming
- Subprograms with local variables
- 8087 math coprocessor supported
- 15 digit floating point arithmetic
- Integer data types
- All math functions operate on arrays
- Matrix operations also included
- Boolean, logical, and binary operators
- Device independent I/O scheme
- Random and Sequential file types
- INPUT statement prompts
- Debug mode with trace options
- Single step function key for manual program execution
- Complete error trapping and handling
- HELP command displays information

Systems Supported

Computers:
IBM PC/XT/AT, HP-150, DEC VAX, Tektronix 4100 Series, and others*

Operating Systems:
MS-DOS, PC-DOS, CPM, UNIX, VMS,...

GPIB Interface Cards:
National Instruments, Ziatech,...

Graphic Cards:
IBM, Hercules, Tecmar...

Graphic Terminals:
Tektronix, Seiko,...

Plotters:
Cal-Comp, Hewlett-Packard, Houston Instruments, Tektronix,...

These and many other popular computers, operating systems, and peripherals are, or will be supported by TBASIC.

*Some of the IBM compatible machines on which TBASIC has been tested are: AT&T 6300, Compaq, IBS, Leading Edge, and Mindset

IBM PC and other micro versions of TBASIC are priced at $795.
A supermini version of TBASIC is available for DEC VAX at $2000.

Call 1-801-224-6550 to order your copy of TBASIC now. Further information and demonstrations are available upon request.

TransEra Corporation
3707 North Canyon Road
Provo, Utah 84604
801-224-6550
Telex: 29-6438

TransEra

CIRCLE NO. 182 ON READER SERVICE CARD

INDEX TO ADVERTISERS

Printer Boss™

Letter quality printing, sideways printing, RAM printer buffer, 5 alternate IBM character sets, graphics, Greek and screen symbols, custom font design, high-speed character-mode output and full control of your Epson printer on one plain-English menu.

Now you can have all of these features and more for your Epson MX, FX, RX, JX and LQ-1500 printers. And all at the touch of a few keys. Printer Boss™ software is simply the ultimate in printer control!

Print letter quality on your FX printer at full text-mode speed using our unique two-pass dot-matrix approach. Use this **high-speed technique** to overprint dots which fit precisely between the dots in the standard characters, filling in the annoying spaces and greatly enhancing readability while avoiding the smearing effect of doublestrike or other print modes. Or print graphics-mode letter quality on the MX, RX and JX printers.

Print multi-page spreadsheets or reports as wide as you like, **sideways on your paper,** in a choice of five different type sizes, with automatic pagination both vertically and horizontally, at high speed and on any Epson printer.

Create a **RAM printer buffer** of from 1 to 32K, so you can print in background while using your PC for other work. Buffer your normal, letter quality or sideways printing for higher efficiency and throughput.

Download any of **5 different IBM character sets,** including the Matrix, Graphics 1 and Graphics 2 printer sets, the Screen set and the APL set. Use IBM block and line graphics characters, IBM screen characters, Greek and APL symbols on any Epson printer. Printer Boss™ is ideal for IBM graphics software and for **screen dumps.**

Or **custom-design your own character font** using our new font design feature, and build a library of your own fonts for future use.

Printer Boss™ also allows **full menu-based operation** of all control functions of all of the Epson printers. Select and deselect pica, elite, compressed, italic, enlarged, emphasized, doublestrike, underline, superscript and subscript for scores of different type faces. Switch the LQ-1500 between draft and letter quality.

Set line spacing, right and left margins and skip-over-perforation. Load USA, France, Germany, England, Denmark, Sweden, Italy, Spain and Japan language sets. **Control** unidirectional, half-speed and proportionally-spaced printing. **Store** and retrieve 10 complete menu settings with a few keystrokes. **Emulate** the IBM-label printer for the IBM-PC.

Print **samples** from the menu, or use program modules provided for printing ASCII tables and spreadsheet and text samples. **Create ASCII files of data,** corresponding to the stored menu settings, which can be accessed by or included in your own programs to set up the printer as you wish.

Printer Boss provides all of the above features on **one easy-to-use menu,** and it's compatible and extremely useful with such word processing and spreadsheet programs as **Wordstar, Lotus 1-2-3, Visicalc** and many, many others. The package is streamlined, professional and intended for serious work by seasoned PC users, but is **easily operable by the neophyte.** As a primary objective the development team designed the menu and its operation to minimize mental effort and fatigue in printer set-up and operation, which is particularly important for the newcomer to computers.

Printer Boss™ runs on the IBM-PC, IBM-PCjr, IBM-XT or any compatibles; PC-DOS 1.1, 2.0, 2.1 or equivalent; double-sided disk drive; 128K RAM; Epson MX, FX, RX, JX or LQ-1500 printers or compatibles.

Printer Boss™ $59.95. With Letter Quality or Sideline™ options $99.95. With both $139.95. **Demo disk** $10.00 credited to order. VISA, MC, or send check. Free info. Dealers welcome, sales kit available.

YES! PLEASE SEND PRINTER BOSS TODAY!

CONFIGURATION:
( ) PRINTER BOSS $59.95
( ) WITH SIDELINE OPTION $99.90
( ) DEMO DISK $10.00
( ) WITH LETTERBOSS OPTION $99.90
( ) WITH BOTH OPTIONS $139.85

ADD $2.00 SHIPPING ALL ORDERS. CONNECTICUT RESIDENTS ADD 7.5%. OUTSIDE USA AND CANADA ADD $15.00. COD ADD $2.00. PURCHASE ORDERS ACCEPTED.

PAYMENT: $_____
( ) CHECK ENCLOSED
( ) MONEY ORDER ENCLOSED
( ) COD
( ) VISA CARD ( ) MASTERCARD
CARD #: _____-_____-_____-_____
EXPIRES: _____
SIGNATURE: _____
DATE: _____

NAME: _____
COMPANY: _____
ADDRESS: _____
CITY/STATE/ZIP: _____
TELEPHONE: _____

Printer Boss™

CONNECTICUT SOFTWARE
30 WILSON AVENUE
ROWAYTON, CT 06853

INFORMATION:
203-838-1844

ORDERS ONLY:
800-826-1589

CIRCLE NO. 235 ON READER SERVICE CARD

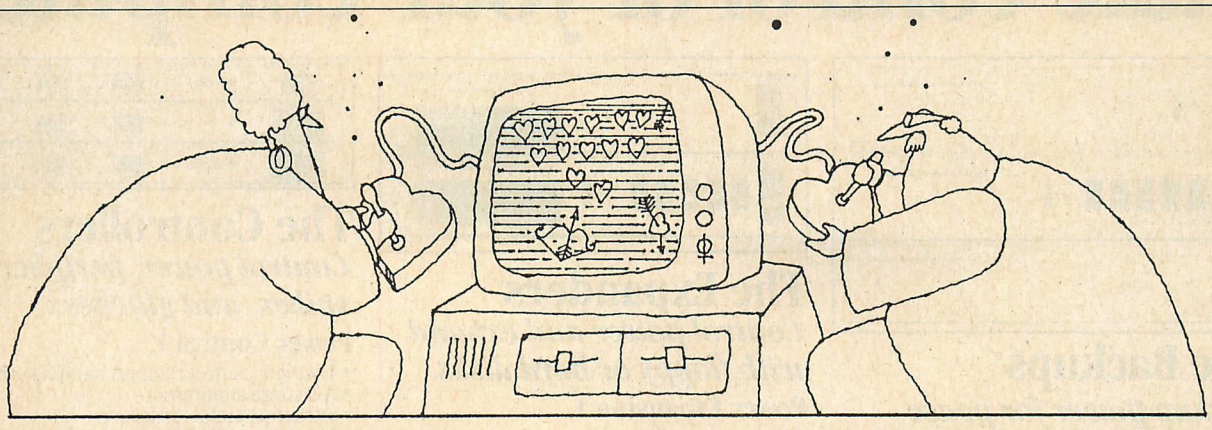


ILLUSTRATION • DAVID POVILAITIS

FEBRUARY

February 4-6
1985 Office Automation Conference
Atlanta, GA
Sponsor: AFIPS
Contact: Trudi Riley, AFIPS, 703/620-8952

February 19-21
Computer Graphics Exhibition and Conference
London, England
Contact: World Computer Graphics Association, Inc., 2033 M Street, N.W., Suite 399, Washington DC, 202/775-9556

February 25-28
Compcon Spring '85, Technological Leverage: A Competitive Necessity
San Francisco, CA
Sponsor: IEEE-CS
Contact: Glen G. Langdon, Jr., IBM Dept. K54/282, 5600 Cottle Road, San Jose, CA 95193, 408/256-6454

MARCH

March 4-7
Interface
Atlanta, GA
Contact: Amy Marks, The Interface Group, 617/449-6600

March 12-14
ACM Computer Science Conference
New Orleans, LA
Sponsor: Association for Computing Machinery
Contact: Della T. Bonnette, Computing and Information Services, University of Southwestern Louisiana, Lafayette, LA 70504; 318/231-6284

March 18-20
COMTEL-85
Dallas, TX
Contact: Michelle Robertson, International Computer and Telecommunications Conference, 214/458-7011

March 20-22
Fourth Annual Phoenix Conference on Computers and Communications
Phoenix, AZ
Sponsor: IEEE-CS
Contact: IEEE PCCC-85, P. O. Box 37125 C, Phoenix, AZ 85069

March 21-24
Comdex/Winter '85
Anaheim, CA
Contact: Amy Marks, The Interface Group, 617/449-6600

March 22-24
3rd Annual Maryland Computer Show & Software Exposition
Baltimore, MD
Sponsor: CompuShows
Contact: CompuShows, Inc., P.O. Box 3315, Annapolis, MD 21403; 800/368-2066

March 25-27
Symposium on Principles of Database Systems
Portland OR
Sponsors: ACM SIGACT and ACM SIGMOD
Contact: Seymour J. Ginsburg, University of Southern California, University Park, Los Angeles, CA 90089-0782; 213/743-5501

March 25-28
IEEE Infocom 85
Arlington, VA
Sponsor: IEEE
Contact: IEEE Infocom 85, P.O. Box 639, Silver Spring, MD 20901; 301/589-8142

March 25-28
1985 IEEE International Conference on Robotics and Automation
St. Louis, MO
Sponsor: IEEE Council on Robotics and Automation
Contact: T. Lozano-Perez, MIT A.I. Lab, 545 Technology Square, Cambridge, MA 02139

March 26-28
Comdex in Japan
Tokyo, Japan
Contact: Milton Berns, The Interface Group, 800/325-3330 or 617/449-6600

March 30-April 2
10th West Coast Computer Faire
San Francisco, CA
Sponsor: Computer Faire, Inc.
Contact: Bill Littlefield, Computer Faire, Inc., 181 Wells Avenue, Suite 201, Newton, MA 02159; 800/826-2680 or 617/965-8350

March 31-April 3
Softcon '85
Atlanta, GA
Contact: Bill Mahan, 800/841-7000 or 617/739-2000

APRIL

April 1-4
SOFTCON/SPRING
New Orleans, LA
Contact: Northeast Exposition, 822 Boylston Street, Chestnut Hill MA 02167; 617/739-2000

April 14-18
Computer Graphics 85—Sixth Annual Conference and Exposition
Dallas, TX
Sponsor: National Computer Graphics Association
Contact: NCGA, 8401 Arlington Blvd., Suite 601, Fairfax, VA 22031; 703/698-9600

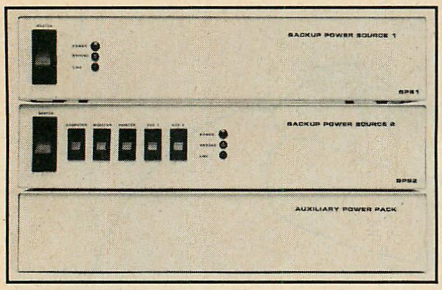
April 14-18
CHI 85: Conference on Human Factors in Computing Systems
San Francisco, CA
Sponsor: ACM SIGCHI
Contact: D. Austin Henderson, Jr., Xerox PARC, 3333 Coyote Hill Road, Palo Alto, CA 94304 or Lorraine Borman, Vogelback Computing Center, Northwestern University, Evanston, IL 60201; 312/492-3682

April 15-17
Conference on Software Tools
New York, NY
Sponsor: Polytechnic Institute of New York in cooperation with ACM and IEEE TCSE
Contact: Martin L. Shooman, Dept. of Electrical Engineering and Computer Science, Polytechnic Institute of New York, 333 Jay Street, New York, NY 11201; 212/643-5174

April 23-25
Federal DP Expo and Conference
Washington, DC
Sponsor: The Interface Group
Contact: Amy Marks, The Interface Group, 617/449-6600

April 23-26
UNIX SYSTEMS EXPO/85
San Francisco, CA
Sponsor: Computer Faire, Inc.
Contact: Sally O'Neill, Computer Faire, Inc., 181 Wells Avenue, Suite 201, Newton, MA 02159, 800/826-2680 or 617/965-8350

Take control of your computer.



The Backups™

Backup power for peace of mind and memory.

Backup Power Source 1

- delivers up to 225 watts @ 120V.AC
- 40 minutes of power at 50% load
- visual and audible power interrupt alarm
- fast automatic switching
  (Within 6 milliseconds of peak voltage detection)
- 2 Outlets
- AC surge suppressor
- 3 stage noise filter
- thermal output protection
- IEC power connector
- attractive metal enclosure
- 3¼" high, 16" wide, 11" deep

$379.95 *complete**

Backup Power Source 2

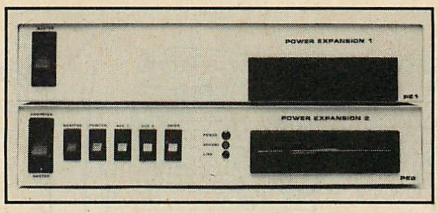
All the features of Power Source 1 plus:
- 6 outlets controlled by front switches
- switchable visual and audible alarm
- 10 amp master switch circuit breaker
- cross suppression between all 6 outlets
- optional 2 hour auxiliary power pack

$499.95 *complete**

Auxiliary Power Pack

- 2 hours of power at full load
- plugs directly into either of our backup units
- includes 6 rechargable 12V power cells
- backup units keep power cells at full charge
- heavy duty power cable
- 3¼" high, 16" wide, 11" deep

$299.95 *complete**



The Expanders™

Control power and expand with floppy or hard disks.

Power Expansion 1

- 1 stage noise filter
- AC surge suppression
- IEC power connector
- attractive metal enclosure
- 3¼" high, 16" wide, 11" deep
- optional internal power supply
- optional floppy and hard disk drives

Power Expansion 1 . . . . . **$149.95***

w/Internal Power Supply **$219.95***

w/Power Supply, DS/DD Drive, All Cables and Instructions. Expands XT or Compatible **$349.95***

w/Power Supply, Half Height 10M Hard Disk Drive, All Cables and Instructions. Expands PC or Compatible . . . **$1149.95***

20M Drive **$1449.95***

Power Expansion 2

All the features of Power Expansion 1 plus:
- 6 outlets controlled by front switches
- 10 amp master switch circuit breaker
- LED ground and line indicators
- 3 stage noise filter
- cross suppression between all 6 outlets
- optional internal power supply
- optional floppy and hard disk drives

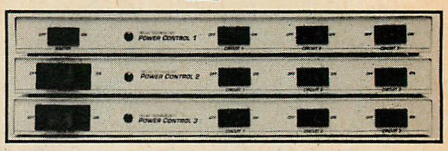
Power Expansion 2 . . . . . **$199.95***

w/Internal Power Supply **$269.95***

w/Power Supply, DS/DD Drive, All Cables and Instructions. Expands XT or Compatible **$399.95***

w/Power Supply, Half Height 10M Hard Disk Drive, All Cables and Instructions. Expands PC or Compatible . . . **$1199.95***

20M Drive **$1499.95***



The Controllers™

Control power, peripherals, spikes, and glitches.

Power Control 1

- 4 outlets control computer and 3 peripherals
- AC surge suppressor
- attractive metal enclosure
- 1¼" high, 16" wide, 10" deep

$69.95 *complete**

Power Control 2

All the features of Power Control 1 plus:
- 10 amp master switch circuit breaker
- 1 stage noise filter
- IEC power connector

$89.95 *complete**

Power Control 3

All the features of Power Control 1 & 2 plus:
- illuminated switches
- 3 stage noise filter
- cross suppression between all 4 outlets

$129.95 *complete**

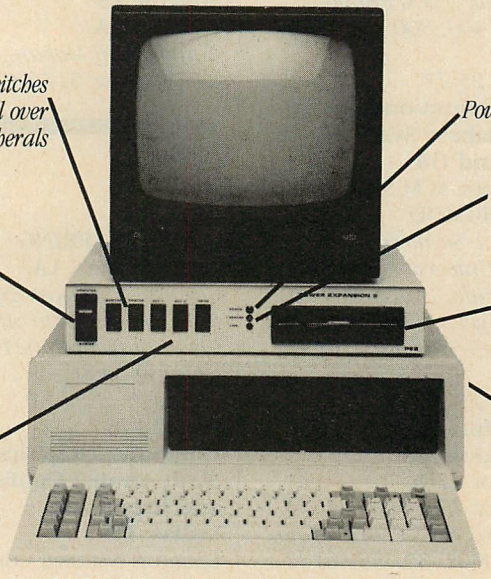
**All prices suggested retail.*

Available at fine computer stores everywhere, or by mail or phone. MasterCard and Visa accepted. *Dealer inquiries invited.*

Relax Technology

The company that works so you can relax and get down to business.

3101 Whipple Rd. #25
Union City, CA 94587
415/471-6112



CIRCLE NO. 193 ON READER SERVICE CARD

You've seen the Macintosh, now see PC Paintbrush

PC Paintbrush

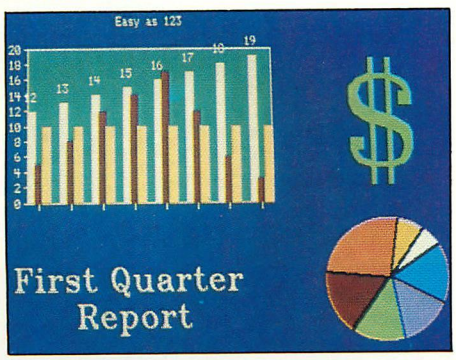
After many months of testing, ZSOFT Corporation is pleased to announce the versatile, colorful, PC Paintbrush.

Not just 2 colors, but 4, 16, or even 256 colors. Not just 320x200 resolution, but 640x200, 640x400, 720x350, 720x704, or more (depending on display adapter).

PC Paintbrush is available NOW for the IBM PC and compatibles, with 192K of memory and DOS 2.0 or above. We support an astonishing variety of hardware; check the lists below. Soon: HP150 and Corona 640x325 versions.

For Business, too

Besides being lots of just plain fun, we've designed PC Paintbrush for serious business. Our character sets (all 6 of them) are the best looking in the industry. Our printout is second to none.



Want to improve humdrum bar graphs? With PC Paintbrush, you can grab a chart or graph from another program (such as Lotus' 1-2-3 or Soreim's SuperCalc3), then embellish it with PC Paintbrush. Take a plain 2 or 4 color bar graph and produce a stunning 16 color slide or printout (requires appropriate display adapter).

We offer special group licensing agreements, as well as inexpensive customization services for your business' special requirements.

Astonishing Resolution

No need to buy additional hardware. We've made sure that PC Paintbrush will work with your printer, your display, and your application.

You can print on B/W printers from IBM, Epson, NEC, Okidata, C-Itoh, and Gemini; or color printers from Diablo, Tektronix, PrintaColor, Hewlett Packard (HP7475 Plotter), Quadram, and Transtar. With the color printers, pick from 25 different colors. With black and white printers, pick from 16 different shading patterns. Color printers never looked so good.

The display possibilities with PC Paintbrush are equally amazing. We COMPLETELY support display cards by Amdek, Quadram, Tecmar, Hercules, IBM, Mindset, Scion, and even the PCjr 16 color display. Just imagine 640x480 resolution in 16 brilliant colors (that's 9 times the resolution of the IBM color card!), or 720x704 resolution in 3 shades of grey.

Blinding Speed

You hate waiting and so do we. All of our graphics are written entirely in assembler, for the fastest possible results.

We provide Products, not Promises

No promises, just fact: we are currently shipping PC Paintbrush.

All ZSOFT software comes with a real warranty, and our service is second to none. Just ask some of our customers (such as IBM, Sohio, Michelin, Salomon Bros., Sperry, Westinghouse, U. of Pittsburgh . . . shall we go on?).

For those interested in updates, a single small fee entitles you to updates for one year; no catch: no additional costs, no need to return diskettes, no postage fees.

Schools

Schools and universities please call for educational discounts.

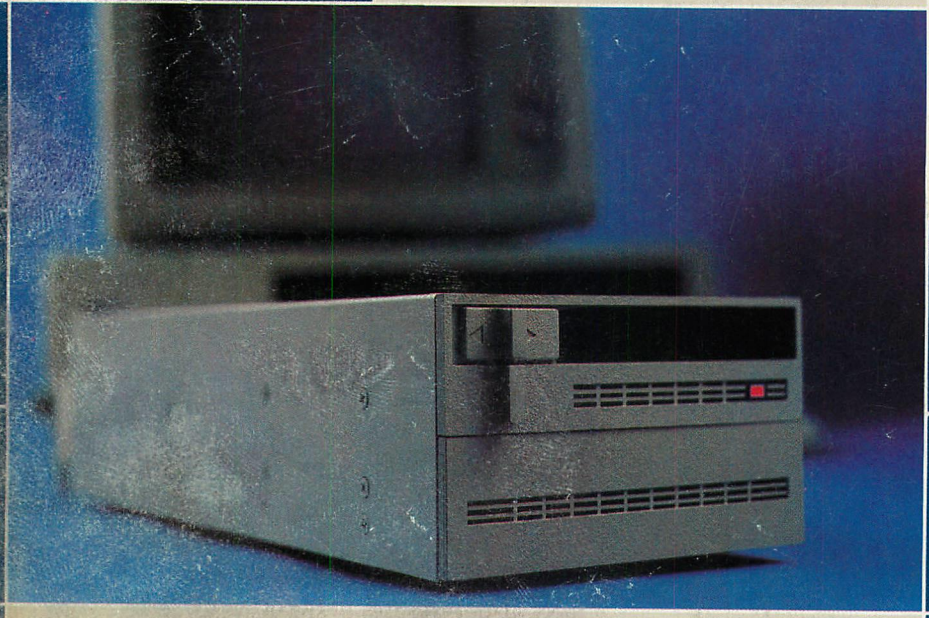
Z·SOFT

370 HERMITAGE CT. SW
MARIETTA, GEORGIA 30064 • 404-425-3766

MacIntosh is a registered trademark of Apple Co.
PCjr. is a registered trademark of IBM.

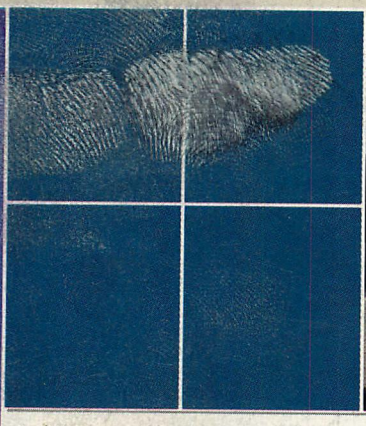
CIRCLE NO. 203 ON READER SERVICE CARD

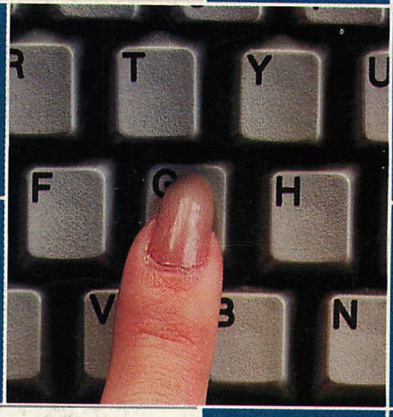
Portable Backup!



Back Up All the Hard Drives in Your Office.

The MaynStream offers fully portable hard drive backup employing the latest software technology. It is compatible with IBM, Compaq, and NCR personal computers* and comes with an industry-leading 1-year warranty.





MAYNSTREAM
TAPE BACKUP SYSTEM
BY MAYNARD ELECTRONICS

Maynard Electronics
430 E. SEMORAN BLVD., CASSELBERRY, FL 32707
305/331-6402

*IBM is a trademark of International Business Machines. Compaq is a trademark of Compaq Computer Corporation. NCR is a trademark of NCR Corporation.

CIRCLE NO. 145 ON READER SERVICE CARD